

SON-2320

PATENT APPLICATION

#4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Patent Application of)
MAKOTO OKA, ET AL) APPLICATION BRANCH
Serial No. To be assigned)
Filed: January 9,2002)
For: PUBLIC KEY CERTIFICATE ISSUING)
SYSTEM, PUBLIC KEY CERTIFICATE)
ISSUING METHOD, DIGITAL)
CERTIFICATION APPARATUS AND)
PROGRAM STORAGE MEDIUM)



CLAIM TO PRIORITY UNDER 35 USC 119

Commissioner for Patents
Washington, D.C. 20231

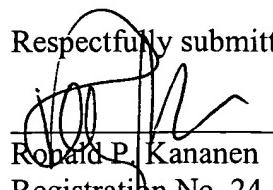
Sir:

The benefit of the filing date of the following prior application filed in the following foreign country is hereby requested and the right of priority provided under 35 U.S.C. 119 is hereby claimed:

Japanese Patent Appl. No. 2001-002220 filed January 10, 2001

In support of this claim, filed herewith is a certified copy of said original foreign application.

Respectfully submitted,


Ronald P. Kananen
Registration No. 24,104

Date: January 8, 2002

RADER, FISHMAN & GRAUER, PLLC
Lion Building
1233 20th Street, N.W.
Washington, D.C. 20036
Tel: (202) 955-37650
Customer No. 23353

日本国特許庁
JAPAN PATENT OFFICE

JCP72 U.S. PRO
10/041964
01/09/02

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出願年月日

Date of Application:

2001年 1月10日

出願番号

Application Number:

特願2001-002220

出願人

Applicant(s):

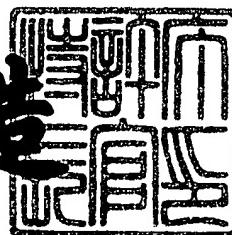
ソニー株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年12月14日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



【書類名】 特許願
【整理番号】 0000579404
【提出日】 平成13年 1月10日
【あて先】 特許庁長官殿
【国際特許分類】 H04L 9/32
【発明者】
【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社
内
【氏名】 岡 誠
【発明者】
【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社
内
【氏名】 石橋 義人
【発明者】
【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社
内
【氏名】 松山 科子
【発明者】
【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社
内
【氏名】 渡辺 秀明
【特許出願人】
【識別番号】 000002185
【氏名又は名称】 ソニー株式会社
【代表者】 出井 伸之
【代理人】
【識別番号】 100101801
【弁理士】
【氏名又は名称】 山田 英治

【電話番号】 03-5541-7577

【選任した代理人】

【識別番号】 100093241

【弁理士】

【氏名又は名称】 宮田 正昭

【電話番号】 03-5541-7577

【選任した代理人】

【識別番号】 100086531

【弁理士】

【氏名又は名称】 澤田 俊夫

【電話番号】 03-5541-7577

【手数料の表示】

【予納台帳番号】 062721

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9904833

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 公開鍵証明書発行システム、公開鍵証明書発行方法、および電子認証装置、並びにプログラム記憶媒体

【特許請求の範囲】

【請求項1】

公開鍵証明書を利用するエンティティの公開鍵証明書を発行する認証局と、管轄エンティティから受領する公開鍵証明書発行要求を前記認証局に対して送信する登録局とを有し、

前記認証局は、各々が異なる署名方式を実行する複数の署名モジュールを有し、前記登録局からの公開鍵証明書発行要求に応じて前記複数の署名モジュールから1以上の署名モジュールを選択し、選択した署名モジュールにおいて公開鍵証明書を構成するメッセージデータに対する電子署名を実行する構成を有することを特徴とする公開鍵証明書発行システム。

【請求項2】

前記認証局は、

複数の署名モジュールと、

前記複数の署名モジュールに対して署名処理要求を出力する認証局サーバを有し、

前記認証局サーバは、前記登録局からの公開鍵証明書発行要求を受信し、該要求に応じて前記複数の署名モジュールから1以上の署名モジュールを選択し、選択した署名モジュールに対して署名処理要求を出力する構成を有し、

前記複数の署名モジュールの各々は、前記認証局サーバから入力した署名処理要求に基づいて公開鍵証明書を構成するメッセージデータに対する電子署名を実行する構成を有することを特徴とする請求項1に記載の公開鍵証明書発行システム。

【請求項3】

前記認証局は、

公開鍵証明書の発行要求を発行する登録局各々と、各登録局に対応して実行すべき署名方式とを対応付けた登録局管理データを格納した登録局管理データベー

スを有し、

登録局からの公開鍵証明書発行要求に従って、前記登録局管理データに基づいて、署名に適用する署名モジュールの選択処理を実行する構成を有することを特徴とする請求項1に記載の公開鍵証明書発行システム。

【請求項4】

前記登録局管理データは、

署名に適用する鍵長、パラメータ情報を含むことを特徴とする請求項3に記載の公開鍵証明書発行システム。

【請求項5】

前記登録局管理データは、

署名に適用する署名モジュールの識別情報を含むことを特徴とする請求項3に記載の公開鍵証明書発行システム。

【請求項6】

前記登録局は、

前記認証局に対する公開鍵証明書の発行要求にともない、署名方式の指定情報を送信し、

前記認証局は、

公開鍵証明書の発行要求に伴って受領する署名方式の指定情報を基づいて、署名に適用する署名モジュールの選択処理を実行する構成を有することを特徴とする請求項1に記載の公開鍵証明書発行システム。

【請求項7】

前記署名方式の指定情報は、

署名に適用する鍵長、パラメータ情報を含むことを特徴とする請求項6に記載の公開鍵証明書発行システム。

【請求項8】

前記認証局は、

前記複数の署名モジュールの各々に対応した署名検証用の検証鍵を格納した検証鍵データベースを有し、

前記複数の署名モジュールの各々の生成した署名の検証処理を実行する構成を

有することを特徴とする請求項1に記載の公開鍵証明書発行システム。

【請求項9】

前記認証局は、

前記複数の署名モジュール中の2以上の署名モジュールを適用して、1つの公開鍵証明書に異なる2以上の電子署名を付加する処理を実行する構成を有することを特徴とする請求項1に記載の公開鍵証明書発行システム。

【請求項10】

前記認証局は、

前記複数の署名モジュール中の2以上の署名モジュールを適用して、各署名モジュールにおいて署名処理の一部ステップを実行し、前記2以上の署名モジュールを連携して適用することにより、電子署名の生成処理を実行する構成を有することを特徴とする請求項1に記載の公開鍵証明書発行システム。

【請求項11】

前記認証局および前記登録局は、

署名方式識別子と、前記複数の署名モジュールの識別子とを対応付けた署名モジュール構成管理テーブルを有し、

登録局は、前記署名モジュール構成管理テーブルに基づいて、署名方式識別子を指定した公開鍵証明書発行要求を認証局に対して発行し、

認証局は、登録局から受領した署名方式識別子に従って、前記署名モジュール構成管理テーブルから対応する署名モジュールの選択を実行する構成を有することを特徴とする請求項1に記載の公開鍵証明書発行システム。

【請求項12】

前記複数の署名モジュールの少なくとも一部の署名モジュールには、共通の署名鍵が格納された構成であることを特徴とする請求項1に記載の公開鍵証明書発行システム。

【請求項13】

前記複数の署名モジュールの各々において実行する署名方式には、複数の署名方式を含むことを特徴とする請求項1に記載の公開鍵証明書発行システム。

【請求項14】

公開鍵証明書を利用するエンティティの公開鍵証明書を発行する認証局と、管轄エンティティから受領する公開鍵証明書発行要求を前記認証局に対して送信する登録局とを有し、登録局からの要求に応じて公開鍵証明書を発行する公開鍵証明書発行方法において、

前記認証局において、

前記登録局からの公開鍵証明書発行要求に応じて、各々が異なる署名方式を実行する複数の署名モジュールから1以上の署名モジュールを選択する署名モジュール選択ステップと、

選択署名モジュールにおいて公開鍵証明書を構成するメッセージデータに対する電子署名を行なう署名ステップと、

を実行することを特徴とする公開鍵証明書発行方法。

【請求項15】

前記公開鍵証明書発行方法は、さらに、

認証局サーバにおいて、前記登録局からの公開鍵証明書発行要求を受信するステップと、

該要求に応じて前記複数の署名モジュールから1以上の署名モジュールを選択するステップと、

選択した署名モジュールに対して署名処理要求を出力するステップと、

を含むことを特徴とする請求項14に記載の公開鍵証明書発行方法。

【請求項16】

前記署名モジュール選択ステップは、

公開鍵証明書の発行要求を発行する登録局各々と、各登録局に対応して実行すべき署名方式とを対応付けた登録局管理データを格納した登録局管理データベースに基づいて選択処理を実行することを特徴とする請求項14に記載の公開鍵証明書発行方法。

【請求項17】

前記署名モジュール選択ステップは、

公開鍵証明書の発行要求に伴って受領する署名方式の指定情報に基づいて、署名に適用する署名モジュールの選択処理を実行することを特徴とする請求項14

に記載の公開鍵証明書発行方法。

【請求項18】

前記公開鍵証明書発行方法は、さらに、

前記認証局において、

前記複数の署名モジュールの各々の生成した署名の検証処理を実行することを特徴とする請求項14に記載の公開鍵証明書発行方法。

【請求項19】

前記公開鍵証明書発行方法において、

前記認証局は、

前記複数の署名モジュール中の2以上の署名モジュールを適用して、1つの公開鍵証明書に異なる2以上の電子署名を付加する処理を実行することを特徴とする請求項14に記載の公開鍵証明書発行方法。

【請求項20】

前記公開鍵証明書発行方法において、

前記認証局は、

前記複数の署名モジュール中の2以上の署名モジュールを適用して、各署名モジュールにおいて署名処理の一部ステップを実行し、前記2以上の署名モジュールを連携して適用することにより、電子署名の生成処理を実行することを特徴とする請求項14に記載の公開鍵証明書発行方法。

【請求項21】

前記公開鍵証明書発行方法において、

前記認証局および前記登録局は、署名方式識別子と、前記複数の署名モジュールの識別子とを対応付けた署名モジュール構成管理テーブルを有し、

登録局は、前記署名モジュール構成管理テーブルに基づいて、署名方式識別子を指定した公開鍵証明書発行要求を認証局に対して発行し、

認証局は、登録局から受領した署名方式識別子に従って、前記署名モジュール構成管理テーブルから対応する署名モジュールの選択を実行することを特徴とする請求項14に記載の公開鍵証明書発行方法。

【請求項22】

前記複数の署名モジュールの各々において実行する署名方式には、複数の署名方式を含むことを特徴とする請求項14に記載の公開鍵証明書発行方法。

【請求項23】

公開鍵証明書を利用するエンティティの公開鍵証明書を発行する認証局を構成する電子認証装置において、

前記電子認証装置は、各々が異なる署名方式を実行する複数の署名モジュールを有し、外部から受信する公開鍵証明書発行要求に応じて前記複数の署名モジュールから1以上の署名モジュールを選択し、選択した署名モジュールにおいて公開鍵証明書を構成するメッセージデータに対する電子署名を実行する構成を有することを特徴とする電子認証装置。

【請求項24】

前記電子認証装置は、

複数の署名モジュールと、

前記複数の署名モジュールに対して署名処理要求を出力する認証局サーバを有し、

前記認証局サーバは、前記公開鍵証明書発行要求を受信し、該要求に応じて前記複数の署名モジュールから1以上の署名モジュールを選択し、選択した署名モジュールに対して署名処理要求を出力する構成を有し、

前記複数の署名モジュールの各々は、前記認証局サーバから入力した署名処理要求に基づいて公開鍵証明書を構成するメッセージデータに対する電子署名を実行する構成を有することを特徴とする請求項23に記載の電子認証装置。

【請求項25】

前記電子認証装置は、

公開鍵証明書の発行要求を発行する登録局各々と、各登録局に対応して実行すべき署名方式とを対応付けた登録局管理データを格納した登録局管理データベースを有し、

登録局からの公開鍵証明書発行要求に従って、前記登録局管理データに基づいて、署名に適用する署名モジュールの選択処理を実行する構成を有することを特徴とする請求項23に記載の電子認証装置。

【請求項26】

前記登録局管理データは、
署名に適用する鍵長、パラメータ情報を含むことを特徴とする請求項25に記載の電子認証装置。

【請求項27】

前記登録局管理データは、
署名に適用する署名モジュールの識別情報を含むことを特徴とする請求項25に記載の電子認証装置。

【請求項28】

前記電子認証装置は、
公開鍵証明書の発行要求に伴って受領する署名方式の指定情報に基づいて、署名に適用する署名モジュールの選択処理を実行する構成を有することを特徴とする請求項23に記載の電子認証装置。

【請求項29】

前記署名方式の指定情報は、
署名に適用する鍵長、パラメータ情報を含むことを特徴とする請求項28に記載の電子認証装置。

【請求項30】

前記電子認証装置は、
前記複数の署名モジュールの各々に対応した署名検証用の検証鍵を格納した検証鍵データベースを有し、
前記複数の署名モジュールの各々の生成した署名の検証処理を実行する構成を有することを特徴とする請求項23に記載の電子認証装置。

【請求項31】

前記電子認証装置は、
前記複数の署名モジュール中の2以上の署名モジュールを適用して、1つの公開鍵証明書に異なる2以上の電子署名を付加する処理を実行する構成を有することを特徴とする請求項23に記載の電子認証装置。

【請求項32】

前記電子認証装置は、

前記複数の署名モジュール中の2以上の署名モジュールを適用して、各署名モジュールにおいて署名処理の一部ステップを実行し、前記2以上の署名モジュールを連携して適用することにより、電子署名の生成処理を実行する構成を有することを特徴とする請求項23に記載の電子認証装置。

【請求項33】

前記電子認証装置は、

署名方式識別子と、前記複数の署名モジュールの識別子とを対応付けた署名モジュール構成管理テーブルを有し、公開鍵証明書発行要求に伴い受領した署名方式識別子に従って、前記署名モジュール構成管理テーブルから対応する署名モジュールの選択を実行する構成を有することを特徴とする請求項23に記載の電子認証装置。

【請求項34】

前記複数の署名モジュールの少なくとも一部の署名モジュールには、共通の署名鍵が格納された構成であることを特徴とする請求項23に記載の電子認証装置。

【請求項35】

前記複数の署名モジュールの各々において実行する署名方式には、複数の署名方式を含むことを特徴とする請求項23に記載の電子認証装置。

【請求項36】

公開鍵証明書を利用するエンティティの公開鍵証明書を発行する公開鍵証明書発行処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム記憶媒体であって、前記コンピュータ・プログラムは、

公開鍵証明書発行要求に応じて、各々が異なる署名方式を実行する複数の署名モジュールから1以上の署名モジュールを選択する署名モジュール選択ステップと、

選択署名モジュールにおいて公開鍵証明書を構成するメッセージデータに対する電子署名を行なう署名ステップと、

を有することを特徴とするプログラム記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は電子配信システムにおいて暗号化データ送信に使用される公開鍵の正当性を証明するための公開鍵証明書の発行処理に関する公開鍵証明書発行システム、公開鍵証明書発行方法、および電子認証装置、並びにプログラム記憶媒体に関する。さらに、公開鍵証明書を発行する認証局（CA）において、複数の署名アルゴリズムに対応した公開鍵証明書を発行し、公開鍵証明書を利用するエンティティにおける利便性を高めた公開鍵証明書発行システム、公開鍵証明書発行方法、および電子認証装置、並びにプログラム記憶媒体に関する。

【0002】

【従来の技術】

昨今、ゲームプログラム、音声データ、画像データ、文書作成プログラム等、様々なソフトウェアデータ（以下、これらをコンテンツ（Content）と呼ぶ）が、インターネット等のネットワークを介して流通している。また、オンラインショッピング等、ネットワークを介した商品売買も次第に盛んになってきている。

【0003】

このようなネットワークを介したデータ通信においては、データ送信側とデータ受信側とが互いに正規なデータ送受信対象であることを確認した上で、必要な情報を転送する、すなわちセキュリティを考慮したデータ転送構成をとるのが一般的となっている。データ転送の際のセキュリティ構成を実現する1つの手法が、転送データの暗号化処理、データに対する署名処理である。

【0004】

暗号化データは、所定の手続きによる復号化処理によって利用可能な復号データ（平文）に戻すことができる。このような情報の暗号化処理に暗号化鍵を用い、復号化処理に復号化鍵を用いるデータ暗号化、復号化方法は従来からよく知られている。

【0005】

暗号化鍵と復号化鍵を用いるデータ暗号化・復号化方法の態様には様々な種類

あるが、その1つの例としていわゆる公開鍵暗号方式と呼ばれる方式がある。公開鍵暗号方式は、発信者と受信者の鍵を異なるものとして、一方の鍵を不特定のユーザが使用可能な公開鍵として、他方を秘密に保つ秘密鍵とするものである。例えば、データ暗号化鍵を公開鍵とし、復号鍵を秘密鍵とする。あるいは、認証子生成鍵を秘密鍵とし、認証子復号鍵を公開鍵とする等の態様において使用される。

【0006】

暗号化、復号化に共通の鍵を用いるいわゆる共通鍵暗号化方式と異なり、公開鍵暗号方式では秘密に保つ必要のある秘密鍵は、特定の1人が持てばよいため鍵の管理において有利である。ただし、公開鍵暗号方式は共通鍵暗号化方式に比較してデータ処理速度が遅く、秘密鍵の配送、デジタル署名等のデータ量の少ない対象に多く用いられている。公開鍵暗号方式の代表的なものにはRSA(Rivest-Shamir-Adleman)暗号がある。これは非常に大きな2つの素数（例えば150桁）の積を用いるものであり、大きな2つの素数（例えば150桁）の積の素因数分解（および離散対数）する処理の困難さを利用している。

【0007】

さらに、公開鍵暗号方式の代表的なものとして、楕円曲線暗号(Elliptic Curve Cryptography(ECC))を用いた方法がある。これは楕円曲線とよばれる曲線上の点の間で演算が定義でき、その上で、離散対数問題の類似物（楕円離散対数問題）が作成可能なことに基づいた方式である。

【0008】

素因数分解問題に基づくRSA暗号方式は、準指數的な解読法を持つのに対して楕円離散対数は、指數的解読しか不可能とされており、素因数分解問題に基づくRSA暗号方式の鍵サイズが512, 1024, または2048ビットとなるのに対して、楕円曲線暗号(ECC)を用いた方式、例えば楕円方式DSA署名(ECDSA)の鍵サイズは160, 192, または224ビット程度で同等の安全性が保持され、鍵サイズの短さによって処理速度を高めることが可能となる。

【0009】

公開鍵暗号方式では、不特定多数に公開鍵を使用可能とする構成であり、配布する公開鍵が正当なものであるか否かを証明する証明書、いわゆる公開鍵証明書を使用する方法が多く用いられている。例えば、利用者Aが公開鍵、秘密鍵のペアを生成して、生成した公開鍵を認証局に対して送付して公開鍵証明書を認証局から入手する。利用者Aは公開鍵証明書を一般に公開する。不特定のユーザは公開鍵証明書から所定の手続きを経て公開鍵を入手して文書等を暗号化して利用者Aに送付する。利用者Aは秘密鍵を用いて暗号化文書等を復号する等のシステムである。また、利用者Aは、秘密鍵を用いて文書等に署名を付け、不特定のユーザが公開鍵証明書から所定の手続きを経て公開鍵を入手して、その署名の検証を行なうシステムである。

【0010】

公開鍵証明書について図1を用いて説明する。公開鍵証明書は、公開鍵暗号方式における認証局（CA : Certification AuthorityまたはIA : Issuing Authority）が発行する証明書であり、ユーザが自己のID、公開鍵等を認証局に提出することにより、認証局側が認証局のIDや有効期限等の情報を付加し、さらに認証局による署名を付加して作成される証明書である。

【0011】

図1に示す公開鍵証明書は、証明書のバージョン番号、認証局（IA）が証明書利用者に対し割り付ける証明書の通し番号、上述のRSA、ECDSA等の電子署名に用いたアルゴリズム、およびパラメータ、認証局の名前、証明書の有効期限、証明書利用者の名前（ユーザID）、証明書利用者の公開鍵並びに電子署名を含む。

【0012】

電子署名は、証明書のバージョン番号、認証局が証明書利用者に対し割り付ける証明書の通し番号、電子署名に用いたアルゴリズムおよびパラメータ、認証局の名前、証明書の有効期限、証明書利用者の名前並びに証明書利用者の公開鍵全体に対し生成される電子署名であり、例えばハッシュ関数を適用してハッシュ値を生成し、そのハッシュ値に対して認証局の秘密鍵を用いて生成したデータである。

【0013】

認証局は、図1に示す公開鍵証明書を発行するとともに、有効期限が切れた公開鍵証明書を更新し、不正を行った利用者の排斥を行うための不正者リストの作成、管理、配布（これをリボケーション：Revocationと呼ぶ）を行う。また、必要に応じて公開鍵・秘密鍵の生成も行う。

【0014】

一方、この公開鍵証明書を利用する際には、利用者は自己が保持する認証局の公開鍵を用い、当該公開鍵証明書の電子署名を検証し、電子署名の検証に成功した後に公開鍵証明書から公開鍵を取り出し、当該公開鍵を利用する。従って、公開鍵証明書を利用する全ての利用者は、共通の認証局の公開鍵を保持している必要がある。

【0015】

【発明が解決しようとする課題】

上述のような認証局発行の公開鍵証明書を用いた公開鍵暗号方式によるデータ送信システムにおいて、公開鍵証明書の電子署名を検証し、電子署名の検証に成功した後に公開鍵証明書から公開鍵を取り出して、例えば公開鍵暗号方式による認証処理、あるいは公開鍵暗号方式による転送データの暗号化処理、あるいは復号処理を実行することが可能となる。しかしながら、公開鍵暗号方式による様々な処理を実行するユーザデバイス等のエンティティは、前述したE C D S A、R S A等の様々な暗号方式アルゴリズムのすべてに対応可能であることは少なく、E C D S Aアルゴリズム、あるいはR S Aアルゴリズムにのみ対応した処理が可能であるといった構成が多い。

【0016】

このような単独、あるいは特定の暗号アルゴリズムのみ処理可能なデバイスは、そのアルゴリズムに従った署名方式を持つ公開鍵証明書のみ利用可能となり、他の方式で署名された公開鍵証明書は受け取っても署名の検証が実施できず、公開鍵証明書の検証が不可能になるという事態が発生する。

【0017】

また、認証局（C A）において署名鍵としての秘密鍵の保持方法や署名付けの

セキュリティの確保は認証局（CA）構築の際の一つの課題であり、さらに署名演算速度の向上も認証局のシステム性能の向上のために必要である。セキュリティ確保、演算速度向上の解決策の1つとして署名鍵（秘密鍵）の保持、署名付けを専用ハードウェア（HSM : Hardware Security Module）で行う構成がある。HSMは高い耐タンパー性を持つため、セキュリティレベル向上にも大きな役割を果たす。現在、HSMを利用してシステムを構築しているケースはあるが、異なる署名アルゴリズムを適用可能とした構成はなく、異なる署名方式に対応した認証局（CA）の構築が望まれる。

【0018】

従来は、例えば図2に示すように、ECDSAアルゴリズムを処理可能なECDSAデバイス23は、ECDSAアルゴリズムによる署名処理を実行するECDSA登録局（ECDSA-RA : Registration Authority）22に公開鍵証明書発行要求、あるいは更新要求を行ない、ECDSA登録局22は、各サービスに参加するエンティティ、機器を認証し、各エンティティ、機器からの公開鍵の公開鍵証明書発行要求を受領し、これをECDSAアルゴリズムによる署名処理を実行するECDSA認証局（ECDSA-CA）21に送信し、ECDSA認証局（ECDSA-CA）21は、ECDSAアルゴリズムによる署名処理を実行した公開鍵証明書を発行してECDSA登録局22を介してECDSAデバイス23に配布する。

【0019】

一方、RSAアルゴリズムを処理可能なRSAデバイス33は、RSAアルゴリズムによる署名処理を実行するRSA登録局（RSA-RA : Registration Authority）32に公開鍵証明書発行要求、あるいは更新要求を行ない、RSA登録局32は、各サービスに参加するエンティティ、機器を認証し、各エンティティ、機器からの公開鍵の公開鍵証明書発行要求を受領し、これをRSAアルゴリズムによる署名処理を実行するRSA認証局（RSA-CA）31に送信し、RSA認証局（RSA-CA）31は、RSAアルゴリズムによる署名処理を実行した公開鍵証明書を発行してRSA登録局32を介してRSAデバイス33に配布する。

【0020】

このように、異なる複数の署名方式に対応したそれぞれの処理系統を構築し、それぞれの処理系統によって構築されたシステム内で閉じられた公開鍵暗号方式による認証、暗号化データ通信が実行される。

【0021】

ECD SAデバイス23は、RSAデバイス33からRSA方式で署名が施されたRSAデバイス33の公開鍵証明書を受信しても署名検証を実行することができず、公開鍵証明書の正当性の検証が不可能になり、証明書としての機能を果たさない。またその逆にRSAデバイス33は、ECD SAデバイス23からECD SA方式で署名が施されたECD SAデバイス23の公開鍵証明書を受信しても署名検証を実行することができず、公開鍵証明書の正当性の検証が不可能になる。

【0022】

図2のECD SAデバイス23と、RSAデバイス33との間でそれぞれ相手方の公開鍵証明書の正当性の確認を行なうためには、それが相手方から受信した公開鍵証明書をECD SA登録局22、RSA登録局32に送信し、さらに、ECD SA認証局(ECD SA-CA)21、RSA認証局(RSA-CA)31に送信し、ECD SA認証局(ECD SA-CA)21、RSA認証局(RSA-CA)31との間で相互に問い合わせを実行し、その結果を各デバイスが受け取って認証の代わりとする方法をとらざる得ない。

【0023】

本発明は、このような公開鍵証明書を用いた公開鍵暗号方式によるデータ通信システムにおける問題点を解決することを目的とするものであり、認証局において、複数の暗号化アルゴリズムをサポートし、複数の暗号化アルゴリズムに対応した公開鍵証明書を発行することにより、ECD SAデバイス、RSAデバイス等、特定の暗号化アルゴリズムのみ処理可能なデバイスのいずれにおいても利用可能となるように、複数の署名方式の署名を付加した公開鍵証明書を単独の認証局において発行し、異なる署名アルゴリズムを処理するデバイス相互間の認証、データ通信において公開鍵証明書の有効利用を可能とした公開鍵証明書発行シス

テム、公開鍵証明書発行方法、および電子認証装置、並びにプログラム記憶媒体を提供することを目的とする。

【0024】

【課題を解決するための手段】

本発明の第1の側面は、

公開鍵証明書を利用するエンティティの公開鍵証明書を発行する認証局と、管轄エンティティから受領する公開鍵証明書発行要求を前記認証局に対して送信する登録局とを有し、

前記認証局は、各々が異なる署名方式を実行する複数の署名モジュールを有し、前記登録局からの公開鍵証明書発行要求に応じて前記複数の署名モジュールから1以上の署名モジュールを選択し、選択した署名モジュールにおいて公開鍵証明書を構成するメッセージデータに対する電子署名を実行する構成を有することを特徴とする公開鍵証明書発行システムにある。

【0025】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記認証局は、複数の署名モジュールと、前記複数の署名モジュールに対して署名処理要求を出力する認証局サーバを有し、前記認証局サーバは、前記登録局からの公開鍵証明書発行要求を受信し、該要求に応じて前記複数の署名モジュールから1以上の署名モジュールを選択し、選択した署名モジュールに対して署名処理要求を出力する構成を有し、前記複数の署名モジュールの各々は、前記認証局サーバから入力した署名処理要求に基づいて公開鍵証明書を構成するメッセージデータに対する電子署名を実行する構成を有することを特徴とする。

【0026】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記認証局は、公開鍵証明書の発行要求を発行する登録局各々と、各登録局に対応して実行すべき署名方式とを対応付けた登録局管理データを格納した登録局管理データベースを有し、登録局からの公開鍵証明書発行要求に従って、前記登録局管理データに基づいて、署名に適用する署名モジュールの選択処理を実行する構成を有することを特徴とする。

【0027】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記登録局管理データは、署名に適用する鍵長、パラメータ情報を含むことを特徴とする。

【0028】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記登録局管理データは、署名に適用する署名モジュールの識別情報を含むことを特徴とする。

【0029】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記登録局は、前記認証局に対する公開鍵証明書の発行要求にともない、署名方式の指定情報を送信し、前記認証局は、公開鍵証明書の発行要求に伴って受領する署名方式の指定情報に基づいて、署名に適用する署名モジュールの選択処理を実行する構成を有することを特徴とする。

【0030】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記署名方式の指定情報は、署名に適用する鍵長、パラメータ情報を含むことを特徴とする。

【0031】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記認証局は、前記複数の署名モジュールの各々に対応した署名検証用の検証鍵を格納した検証鍵データベースを有し、前記複数の署名モジュールの各々の生成した署名の検証処理を実行する構成を有することを特徴とする。

【0032】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記認証局は、前記複数の署名モジュール中の2以上の署名モジュールを適用して、1つの公開鍵証明書に異なる2以上の電子署名を付加する処理を実行する構成を有することを特徴とする。

【0033】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記認証局は、前記複数の署名モジュール中の2以上の署名モジュールを適用して、各署名モジュールにおいて署名処理の一部ステップを実行し、前記2以上の署名モジュールを連携して適用することにより、電子署名の生成処理を実行する構成を有することを特徴とする。

【0034】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記認証局および前記登録局は、署名方式識別子と、前記複数の署名モジュールの識別子とを対応付けた署名モジュール構成管理テーブルを有し、登録局は、前記署名モジュール構成管理テーブルに基づいて、署名方式識別子を指定した公開鍵証明書発行要求を認証局に対して発行し、認証局は、登録局から受領した署名方式識別子に従って、前記署名モジュール構成管理テーブルから対応する署名モジュールの選択を実行する構成を有することを特徴とする。

【0035】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記複数の署名モジュールの少なくとも一部の署名モジュールには、共通の署名鍵が格納された構成であることを特徴とする。

【0036】

さらに、本発明の公開鍵証明書発行システムの一実施態様において、前記複数の署名モジュールの各々において実行する署名方式には、複数の署名方式を含むことを特徴とする。

【0037】

さらに、本発明の第2の側面は、

公開鍵証明書を利用するエンティティの公開鍵証明書を発行する認証局と、管轄エンティティから受領する公開鍵証明書発行要求を前記認証局に対して送信する登録局とを有し、登録局からの要求に応じて公開鍵証明書を発行する公開鍵証明書発行方法において、

前記認証局において、

前記登録局からの公開鍵証明書発行要求に応じて、各々が異なる署名方式を実

行する複数の署名モジュールから 1 以上の署名モジュールを選択する署名モジュール選択ステップと、

選択署名モジュールにおいて公開鍵証明書を構成するメッセージデータに対する電子署名を行なう署名ステップと、

を実行することを特徴とする公開鍵証明書発行方法にある。

【0038】

さらに、本発明の公開鍵証明書発行方法の一実施態様において、認証局サーバにおいて、前記登録局からの公開鍵証明書発行要求を受信するステップと、該要求に応じて前記複数の署名モジュールから 1 以上の署名モジュールを選択するステップと、選択した署名モジュールに対して署名処理要求を出力するステップとを含むことを特徴とする。

【0039】

さらに、本発明の公開鍵証明書発行方法の一実施態様において、前記署名モジュール選択ステップは、公開鍵証明書の発行要求を発行する登録局各々と、各登録局に対応して実行すべき署名方式とを対応付けた登録局管理データを格納した登録局管理データベースに基づいて選択処理を実行することを特徴とする。

【0040】

さらに、本発明の公開鍵証明書発行方法の一実施態様において、前記署名モジュール選択ステップは、公開鍵証明書の発行要求に伴って受領する署名方式の指定情報に基づいて、署名に適用する署名モジュールの選択処理を実行することを特徴とする。

【0041】

さらに、本発明の公開鍵証明書発行方法の一実施態様において、前記認証局において、前記複数の署名モジュールの各々の生成した署名の検証処理を実行することを特徴とする。

【0042】

さらに、本発明の公開鍵証明書発行方法の一実施態様において、前記認証局は、前記複数の署名モジュール中の 2 以上の署名モジュールを適用して、1 つの公開鍵証明書に異なる 2 以上の電子署名を付加する処理を実行することを特徴とす

る。

【0043】

さらに、本発明の公開鍵証明書発行方法の一実施態様において、前記認証局は、前記複数の署名モジュール中の2以上の署名モジュールを適用して、各署名モジュールにおいて署名処理の一部ステップを実行し、前記2以上の署名モジュールを連携して適用することにより、電子署名の生成処理を実行することを特徴とする。

【0044】

さらに、本発明の公開鍵証明書発行方法の一実施態様において、前記認証局および前記登録局は、署名方式識別子と、前記複数の署名モジュールの識別子とを対応付けた署名モジュール構成管理テーブルを有し、登録局は、前記署名モジュール構成管理テーブルに基づいて、署名方式識別子を指定した公開鍵証明書発行要求を認証局に対して発行し、認証局は、登録局から受領した署名方式識別子に従って、前記署名モジュール構成管理テーブルから対応する署名モジュールの選択を実行することを特徴とする。

【0045】

さらに、本発明の公開鍵証明書発行方法の一実施態様において、前記複数の署名モジュールの各々において実行する署名方式には、複数の署名方式を含むことを特徴とする。

【0046】

さらに、本発明の第3の側面は、

公開鍵証明書を利用するエンティティの公開鍵証明書を発行する認証局を構成する電子認証装置において、

前記電子認証装置は、各々が異なる署名方式を実行する複数の署名モジュールを有し、外部から受信する公開鍵証明書発行要求に応じて前記複数の署名モジュールから1以上の署名モジュールを選択し、選択した署名モジュールにおいて公開鍵証明書を構成するメッセージデータに対する電子署名を実行する構成を有することを特徴とする電子認証装置にある。

【0047】

さらに、本発明の電子認証装置の一実施態様において、前記電子認証装置は、複数の署名モジュールと、前記複数の署名モジュールに対して署名処理要求を出力する認証局サーバを有し、前記認証局サーバは、前記公開鍵証明書発行要求を受信し、該要求に応じて前記複数の署名モジュールから1以上の署名モジュールを選択し、選択した署名モジュールに対して署名処理要求を出力する構成を有し、前記複数の署名モジュールの各々は、前記認証局サーバから入力した署名処理要求に基づいて公開鍵証明書を構成するメッセージデータに対する電子署名を実行する構成を有することを特徴とする。

【0048】

さらに、本発明の電子認証装置の一実施態様において、前記電子認証装置は、公開鍵証明書の発行要求を発行する登録局各々と、各登録局に対応して実行すべき署名方式とを対応付けた登録局管理データを格納した登録局管理データベースを有し、登録局からの公開鍵証明書発行要求に従って、前記登録局管理データに基づいて、署名に適用する署名モジュールの選択処理を実行する構成を有することを特徴とする。

【0049】

さらに、本発明の電子認証装置の一実施態様において、前記登録局管理データは、署名に適用する鍵長、パラメータ情報を含むことを特徴とする。

【0050】

さらに、本発明の電子認証装置の一実施態様において、前記登録局管理データは、署名に適用する署名モジュールの識別情報を含むことを特徴とする。

【0051】

さらに、本発明の電子認証装置の一実施態様において、前記電子認証装置は、公開鍵証明書の発行要求に伴って受領する署名方式の指定情報に基づいて、署名に適用する署名モジュールの選択処理を実行する構成を有することを特徴とする。

【0052】

さらに、本発明の電子認証装置の一実施態様において、前記署名方式の指定情報は、署名に適用する鍵長、パラメータ情報を含むことを特徴とする。

【0053】

さらに、本発明の電子認証装置の一実施態様において、前記電子認証装置は、前記複数の署名モジュールの各々に対応した署名検証用の検証鍵を格納した検証鍵データベースを有し、前記複数の署名モジュールの各々の生成した署名の検証処理を実行する構成を有することを特徴とする。

【0054】

さらに、本発明の電子認証装置の一実施態様において、前記電子認証装置は、前記複数の署名モジュール中の2以上の署名モジュールを適用して、1つの公開鍵証明書に異なる2以上の電子署名を付加する処理を実行する構成を有することを特徴とする。

【0055】

さらに、本発明の電子認証装置の一実施態様において、前記電子認証装置は、前記複数の署名モジュール中の2以上の署名モジュールを適用して、各署名モジュールにおいて署名処理の一部ステップを実行し、前記2以上の署名モジュールを連携して適用することにより、電子署名の生成処理を実行する構成を有することを特徴とする。

【0056】

さらに、本発明の電子認証装置の一実施態様において、前記電子認証装置は、署名方式識別子と、前記複数の署名モジュールの識別子とを対応付けた署名モジュール構成管理テーブルを有し、公開鍵証明書発行要求に伴い受領した署名方式識別子に従って、前記署名モジュール構成管理テーブルから対応する署名モジュールの選択を実行する構成を有することを特徴とする。

【0057】

さらに、本発明の電子認証装置の一実施態様において、前記複数の署名モジュールの少なくとも一部の署名モジュールには、共通の署名鍵が格納された構成であることを特徴とする。

【0058】

さらに、本発明の電子認証装置の一実施態様において、前記複数の署名モジュールの各々において実行する署名方式には、複数の署名方式を含むことを特徴と

する。

【0059】

さらに、本発明の第4の側面は、

公開鍵証明書を利用するエンティティの公開鍵証明書を発行する公開鍵証明書発行処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム記憶媒体であって、前記コンピュータ・プログラムは、

公開鍵証明書発行要求に応じて、各々が異なる署名方式を実行する複数の署名モジュールから1以上の署名モジュールを選択する署名モジュール選択ステップと、

選択署名モジュールにおいて公開鍵証明書を構成するメッセージデータに対する電子署名を行なう署名ステップと、

を有することを特徴とするプログラム記憶媒体にある。

【0060】

なお、本発明の第4の側面に係るプログラム記憶媒体は、例えば、様々なプログラム・コードを実行可能な汎用コンピュータ・システムに対して、コンピュータ・プログラムをコンピュータ可読な形式で提供する媒体である。媒体は、CDやFD、MOなどの記録媒体、あるいは、ネットワークなどの伝送媒体など、その形態は特に限定されない。

【0061】

このようなプログラム記憶媒体は、コンピュータ・システム上で所定のコンピュータ・プログラムの機能を実現するための、コンピュータ・プログラムと提供媒体との構造上又は機能上の協働的関係を定義したものである。換言すれば、該提供媒体を介してコンピュータ・プログラムをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協働的作用が發揮され、本発明の他の側面と同様の作用効果を得ることができるのである。

【0062】

本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【0063】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態について詳細に説明する。まず、以下の実施例中で使用する用語の意味について説明する。

認証局（C A : Certification Authority）

公開鍵証明書を作成、発行する機関。

登録局（R A : Registration Authority）

公開鍵証明書を発行するための登録業務を行う。

ユーザ、サービスプロバイダ、サーバなど公開鍵証明書の利用主体であるエンドエンティティ（E E）からの公開鍵証明書発行依頼を受け、認証局（C A）に公開鍵証明書発行要求を行う。認証局（C A）により発行された証明書はエンドエンティティ（E E）に渡される。

ハードウェア・セキュリティ・モジュール（H S M : Hardware Security Module）

署名鍵を保持し、証明書に署名付けを行う専用ハードウェア。

エンドエンティティ（E E : End Entity）

公開鍵証明書発行対象。すなわち公開鍵証明書を利用する機器やサーバ、あるいはユーザ、サービスプロバイダなどである。

【0064】

〔複数の署名モジュールを持つ認証局（C A）〕

まず、複数の署名モジュールを持つ電子認証装置としての認証局（C A）構成について説明する。公開鍵暗号系を用いたシステムにおいて秘密鍵の保持方法や署名付けのセキュリティの確保は認証局（C A : Certificate Authority）を構築する際の一つの課題であり、また、署名付けの演算速度向上は認証局のシステム性能の向上をもたらす。

【0065】

セキュリティ確保、演算速度向上の解決策の1つとして署名鍵（秘密鍵）の保持、署名付けを専用ハードウェア（H S M : Hardware Security Module）で行うことが可能である。H S Mは高い耐タンパー性を持つため、セキュリティレベル向上にも大きな役割を果たす。しかし、専用ハードウェア（H S M）で実行される暗

号化アルゴリズムは固定的なものとなってしまい、署名アルゴリズムを変更して実行する運用は困難である。

【0066】

本発明のシステムにおいては、認証局（CA）は、複数の異なる署名方式、鍵長、パラメータを適用可能とした構成を持つ。具体的にはそれぞれが異なる署名アルゴリズムを実行する専用ハードウェア（HSM）、またはソフトウェアによる複数の署名モジュールを有する認証局（CA）構成を実現する。

【0067】

図3に複数の署名モジュールを持つ電子認証装置としての認証局（CA）の処理を説明する図を示す。認証局（CA）70のCAサーバ71は、公開鍵証明書を利用する機器やサーバ、あるいはユーザ、サービスプロバイダなど公開鍵証明書発行対象エンドエンティティ（EE：End Entity）からの公開鍵証明書発行要求を様々な登録局（RA：Registration Authority）81～85を介して受信する。

【0068】

各登録局（RA）81～85は、自己の管轄するエンドエンティティ（EE）に対して発行する公開鍵証明書に対する署名アルゴリズム、例えば、RSA（Rivest-Shamir-Adleman）暗号方式、あるいは楕円曲線暗号（ECC：Elliptic Curve Cryptography）などを許容する署名方式として規程しており、規程された署名方式から1つまたは複数のアルゴリズムによる署名を実行した公開鍵証明書の発行要求を認証局70に対して実行する。各登録局（RA）81～85は、自己の管轄するエンドエンティティ（EE）において処理可能、すなわち検証処理可能な暗号化アルゴリズムに従った署名がなされた公開鍵証明書の発行を要求することになり、各登録局（RA）81～85ごとに希望する署名アルゴリズムは異なることになる。

【0069】

公開鍵証明書の発行要求は、認証局（CA）70のCAサーバ71によって受け付られ、CAサーバの有する各登録局（RA）81～85と適用署名アルゴリズムの種類を対応付けたテーブルに従って、署名モジュール72a～72nが選

択され、選択された署名モジュールに生成した公開鍵証明書を送信するとともに署名実行命令を送信する。

【0070】

公開鍵証明書と署名実行命令を受信した各署名モジュールは、それぞれのモジュールによって実行可能な署名アルゴリズム（ex. RSA, ECDSA）に従って署名処理を実行し、署名が実行された公開鍵証明書をCAサーバ71に返送する。CAサーバ71は、各モジュールにおいて署名のなされた公開鍵証明書を各モジュールから受信して発行要求を出した登録局（RA）81～85に送付する。

【0071】

署名モジュール72a～72nの各々は、署名を実行するための署名アルゴリズムに従った認証局の署名鍵を外部から入力して格納するかまたは自ら生成し、その署名鍵を用いて署名を実行する。署名モジュール72a～72nの各々は、署名実行用の専用ハードウェア(HSM: Hardware Security Module)、あるいは署名アルゴリズムを実行可能なプログラムによって署名を実行する専用プロセッサ、あるいはCPUを備えたモジュールとして構成される。署名モジュール72a～72nの各々は耐タンパー性を持ち、CAサーバ71の生成した公開鍵証明書の構成要素に基づくメッセージに対して署名鍵による署名処理を実行する。なお、以下の説明では、署名モジュールを持つ処理部をHSMとして説明するが、HSMは、署名アルゴリズムを実行可能なプログラムによって署名を実行する専用プロセッサ、あるいはCPUを備えたソフトウェアによる署名処理を実行するモジュールによっても置き換え可能である。

【0072】

署名モジュール72a～72nの各々の実行する署名処理の例として、RSA(Rivest-Shamir-Adleman)暗号方式、楕円DSA署名方式(ECDSA:Elliptic Curve Digital Signature Algorithm)などがあり、さらに、各方式において適用する鍵長により、演算速度、セキュリティ度合いなどが異なる。鍵長としては、例えば現在使われているものとして、RSA暗号：512bit、1024bit、2048bit、また、ECDSAでは、160bit、192bit

t 、224 bit がある。また、ECDSA方式については、体 $F(p)$ (p は素数または2のべき乗) 上の楕円曲線 $y^2 = x^3 + ax + b$ において、体の標数 p 、位数 r 、 a 、 b 、曲線上のベースポイント Gx, Gy によって署名付けを行うアルゴリズムが決定しセキュリティ強度も異なる。この署名アルゴリズムについては、後段で説明する。

【0073】

[公開鍵証明書]

公開鍵証明書は、公開鍵を用いた暗号データの送受信、あるいはデータ送受信を行なう2者間での相互認証等の処理において、使用する公開鍵が正当な利用者の有する公開鍵であることを第三者、すなわち公開鍵証明書の発行局としての認証局 (CA : Certification Authority) が証明したものである。本発明のシステムで使用される公開鍵証明書の詳細構成について図4、図5を用いて説明する。図4、5の公開鍵証明書のフォーマット例は、公開鍵証明書フォーマットX.509 V3に準拠した例である。

【0074】

バージョン (version) は、証明書フォーマットのバージョンを示す。

シリアルナンバ (Serial Number) は、認証局 (CA) によって設定される公開鍵証明書のシリアルナンバである。

署名アルゴリズム識別子、アルゴリズムパラメータ (Signature algorithm Identifier algorithm parameter) は、公開鍵証明書の署名アルゴリズムとそのパラメータを記録するフィールドである。なお、署名アルゴリズムとしては、楕円曲線暗号 (ECC)、RSAなどがあり、楕円曲線暗号が適用されている場合はパラメータおよび鍵長が記録され、RSAが適用されている場合には鍵長が記録される。さらに、他の暗号方式が適用されればその方式の識別子が記録される。

発行者 (issuer) は、公開鍵証明書の発行者、すなわち認証局 (CA) の名称が識別可能な形式 (Distinguished Name) で記録されるフィールドである。

有効期限 (validity) は、証明書の有効期限である開始日時、終了日時が記録される。

サブジェクト (subject) は、ユーザである認証対象者の名前が記録される。具

体的には例えばユーザ機器のIDや、サービス提供主体のID等である。

サブジェクト公開鍵情報 (subject Public Key Info algorithm subject Public key) は、ユーザの公開鍵情報としての鍵アルゴリズム、鍵情報そのものを格納するフィールドである。

【0075】

ここまでが、公開鍵証明書フォーマットX.509 V1に含まれるフィールドであり、以下は、公開鍵証明書フォーマットX.509 V3において追加されるフィールドである。

【0076】

証明局鍵識別子 (authority Key Identifier-key Identifier, authority Cert Issuer, authority Cert Serial Number) は、認証局 (CA) の鍵を識別する情報であり、鍵識別番号 (8進数)、認証局 (CA) の名称、認証番号を格納する。

サブジェクト鍵識別子 (subject key Identifier) は、複数の鍵を公開鍵証明書において証明する場合に各鍵を識別するための識別子を格納する。

鍵使用目的 (key usage) は、鍵の使用目的を指定するフィールドであり、(0) デジタル署名用、(1) 否認防止用、(2) 鍵の暗号化用、(3) メッセージの暗号化用、(4) 共通鍵配達用、(5) 認証の署名確認用、(6) 失効リストの署名確認用の各使用目的が設定される。

秘密鍵有効期限 (private Key Usage Period) は、ユーザの有する秘密鍵の有効期限を記録する。

認証局ポリシー (certificate Policies) は、認証局 (CA) および登録局 (RA) の証明書発行ポリシーを記録する。例えばISO/IEC 9384-1に準拠したポリシーID、認証基準である。

ポリシー・マッピング (policy Mapping) は、認証局 (CA) を認証する場合にのみ記録するフィールドであり、証明書発行を行なう認証局 (CA) のポリシーと、被認証ポリシーのマッピングを規定する。

サポート・アルゴリズム (supported Algorithms) は、ディレクトリ (X.500) のアトリビュートを定義する。これは、コミュニケーションの相手がディ

レクトリ情報を利用する場合に、事前にそのアトリビュートを知らせるのに用いる。

サブジェクト別名 (subject Alt Name) は、ユーザの別名を記録するフィールドである。

発行者別名 (issuer Alt Name) は、証明書発行者の別名を記録するフィールドである。

サブジェクト・ディレクトリ・アトリビュート (subject Directory Attribute) は、ユーザの任意の属性を記録するフィールドである。

基本制約 (basic Constraint) は、証明対象の公開鍵が認証局 (CA) の署名用か、ユーザのものかを区別するためのフィールドである。

名称制約 (name Constraints) は、被認証者が認証局 (CA) である場合にのみ使用される証明書の有効領域を示すフィールドである。

ポリシー制約 (policy Constraints) は、認証パスの残りに対する明確な認証ポリシー ID、禁止ポリシーマップを要求する制限を記述する。

CRL配布ポイント (Certificate Revocation List Distribution Points) は、ユーザが証明書を利用する際に、証明書が失効していないか、どうかを確認するための失効リストの参照ポイントを記述するフィールドである。

署名は、認証局 (CA) の署名フィールドである。

【0077】

[署名アルゴリズム]

上述の公開鍵証明書の署名は、公開鍵証明書発行局 (CA) の秘密鍵を用いて公開鍵証明書のデータに対して実行される電子署名であり、公開鍵証明書の利用者は、公開鍵証明書発行局 (CA) の公開鍵を用いて検証を行ない、公開鍵証明書の改竄有無がチェック可能となっている。

【0078】

電子署名のアルゴリズムについて、まず、図6を用いて楕円曲線暗号 (Elliptic Curve Cryptography (ECC)) を用いた処理について説明する。図6に示す処理は、ECDSA ((Elliptic Curve Digital Signature Algorithm)、IEEE P1363/D3) を用いた電子署名データの生成処理フローである。

【0079】

図6の各ステップについて説明する。ステップS1において、 p を標数、 a, b を楕円曲線の係数（楕円曲線： $4a^3 + 27b^2 \neq 0 \pmod{p}$ ）、 G を楕円曲線上のベースポイント、 r を G の位数、 K_s を秘密鍵（ $0 < K_s < r$ ）とする。ステップS2において、メッセージMのハッシュ値を計算し、 $f = \text{Hash}(M)$ とする。

【0080】

ここで、ハッシュ関数を用いてハッシュ値を求める方法を説明する。ハッシュ関数とは、メッセージを入力とし、これを所定のビット長のデータに圧縮し、ハッシュ値として出力する関数である。ハッシュ関数は、ハッシュ値（出力）から入力を予測することが難しく、ハッシュ関数に入力されたデータの1ビットが変化したとき、ハッシュ値の多くのビットが変化し、また、同一のハッシュ値を持つ異なる入力データを探し出すことが困難である特徴を有する。ハッシュ関数としては、MD4、MD5、SHA-1などが用いられる場合もあるし、DES-CBCが用いられる場合もある。この場合は、最終出力値となるMAC（チェック値：ICVに相当する）がハッシュ値となる。

【0081】

続けて、ステップS3で、乱数 u （ $0 < u < r$ ）を生成し、ステップS4でベースポイントを u 倍した座標 V （ X_v, Y_v ）を計算する。なお、楕円曲線上の加算、2倍算は次のように定義されている。

【0082】

【数1】

$P = (X_a, Y_a), Q = (X_b, Y_b), R = (X_c, Y_c) = P+Q$ とすると、

$P \neq Q$ の時（加算）、

$$X_c = \lambda^2 - X_a - X_b$$

$$Y_c = \lambda \times (X_a - X_c) - Y_a$$

$$\lambda = (Y_b - Y_a) / (X_b - X_a)$$

$P = Q$ の時（2倍算）、

$$X_c = \lambda^2 - 2X_a$$

$$Y_c = \lambda \times (X_a - X_c) - Y_a$$

$$\lambda = (3(Xa)^2 + a) / (2Ya)$$

【0083】

これらを用いて点Gのu倍を計算する（速度は遅いが、最もわかりやすい演算方法として次のように行う。G、 $2 \times G$ 、 $4 \times G \dots$ を計算し、uを2進数展開して1が立っているところに対応する $2^i \times G$ （Gをi回2倍算した値（iはuのLSBから数えた時のビット位置））を加算する。

【0084】

ステップS5で、 $c = X v \bmod r$ を計算し、ステップS6でこの値が0になるかどうか判定し、0でなければステップS7で $d = [(f + cKs) / u]$
 $\bmod r$ を計算し、ステップS8でdが0であるかどうか判定し、dが0でなければ、ステップS9でcおよびdを電子署名データとして出力する。仮に、rを160ビット長の長さであると仮定すると、電子署名データは320ビット長となる。

【0085】

ステップS6において、cが0であった場合、ステップS3に戻って新たな乱数を生成し直す。同様に、ステップS8でdが0であった場合も、ステップS3に戻って乱数を生成し直す。

【0086】

次に、ECCによる電子署名の検証方法を、図7を用いて説明する。ステップS11で、Mをメッセージ、pを標数、a、bを楕円曲線の係数（楕円曲線： $y^2 = x^3 + ax + b$ ）、Gを楕円曲線上のベースポイント、rをGの位数、GおよびKs×Gを公開鍵（ $0 < Ks < r$ ）とする。ステップS12で電子署名データcおよびdが $0 < c < r$ 、 $0 < d < r$ を満たすか検証する。これを満たしていた場合、ステップS13で、メッセージMのハッシュ値を計算し、 $f = \text{Hash}(M)$ とする。次に、ステップS14で $h_1 = 1/d \bmod r$ を計算し、ステップS15で $h_1 = f h \bmod r$ 、 $h_2 = ch \bmod r$ を計算する。

【0087】

ステップS16において、既に計算した h_1 および h_2 を用い、点 $P = (X_p, Y_p) = h_1 \times G + h_2 \cdot Ks \times G$ を計算する。電子署名検証者は、公開鍵G

および $K s \times G$ を知っているので、図6のステップS4と同様に橍円曲線上の点のスカラー倍の計算ができる。そして、ステップS17で点Pが無限遠点かどうか判定し、無限遠点でなければステップS18に進む（実際には、無限遠点の判定はステップS16でできてしまう。つまり、 $P = (X, Y)$ 、 $Q = (X, -Y)$ の加算を行うと、 λ が計算できず、 $P + Q$ が無限遠点であることが判明している）。ステップS18で $X_p \bmod r$ を計算し、電子署名データcと比較する。最後に、この値が一致していた場合、ステップS19に進み、電子署名が正しいと判定する。

【0088】

電子署名が正しいと判定された場合、データは改竄されておらず、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したことがわかる。

【0089】

ステップS12において、電子署名データcまたはdが、 $0 < c < r$ 、 $0 < d < r$ を満たさなかった場合、ステップS20に進む。また、ステップS17において、点Pが無限遠点であった場合もステップS20に進む。さらにまた、ステップS18において、 $X_p \bmod r$ の値が、電子署名データcと一致していなかった場合にもステップS20に進む。

【0090】

ステップS20において、電子署名が正しくないと判定された場合、データは改竄されているか、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したのではないことがわかる。

【0091】

次に、RSA暗号方式の署名アルゴリズムを図8、図9を用いて説明する。図8は、RSA暗号方式の署名および署名検証用の公開鍵、秘密鍵の生成方法を示し、図9は（a）署名生成処理、（b）署名検証処理を示している。

【0092】

図8の署名および署名検証用の公開鍵、秘密鍵の生成フローにおいて、まず素数p、q（150桁程度）を選択し（S21）、 $n = p \cdot q$ を計算し（S22）、さらに、 $L = (p - 1) \cdot (q - 1)$ を計算し（S23）、Lと共通因数を持たな

い n 未満の正整数 e を選択し、 (n, e) を公開鍵とし (S 24) 、 $d \equiv 1 \pmod L$ を満足する正整数 d を求めて (p, q, d) を秘密鍵とする (S 25)。

【0093】

このような公開鍵、秘密鍵を用いた署名の生成、検証は図9のフローに従って行われる。図9 (a) に示す署名生成は、ステップ S 3 1において、署名対象となるメッセージ M に対してハッシュ間数 h を適用して $m = h(M)$ を生成し、さらに、 $S = m^d \pmod n$ を生成 (S 3 2) して、 S を署名とする。

【0094】

図9 (b) に示す署名検証は、署名検証対象となるメッセージ M に対してハッシュ間数 h を適用して $m = h(M)$ を生成 (S 3 3) し、さらに、 $m = S^e \pmod d$ が成立するか否かを検証 (S 3 4) して、成立する場合は、署名が正しい (S 3 5) と判定する。

【0095】

署名が正しいと判定された場合、データは改竄されておらず、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したことがわかる。

【0096】

ステップ S 3 4において、 $m = S^e \pmod n$ が成立しないと判定された場合、ステップ S 3 6において署名が間違っていると判定され、データは改竄されているか、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したのではないことがわかる。

【0097】

このように、公開鍵証明書の署名検証により、公開鍵証明所の正当性が検証されることになる。上述したように、署名の検証は、署名アルゴリズムに従った暗号処理を実行することが必要であり、エンドエンティティの機器において実行可能な署名アルゴリズムであることが必要であり、一般には登録局 (RA) の管理の下に共通の署名アルゴリズムが用いられる。

【0098】

[認証局 (CA) 構成]

本発明のシステムにおいて、上述の公開鍵証明書を発行する機関としての認証局（CA）、すなわち電子認証装置は、上述のRSA、ECDSAを含む様々な署名アルゴリズムを実行可能な構成であり、前述したように複数の署名モジュールを持つ。認証局（CA）は、複数の異なる署名方式、パラメータを適用可能とした構成を持ち、RSA（Rivest-Shamir-Adleman）暗号方式、ECDSA署名方式（ECDSA : Elliptic Curve Digital Signature Algorithm）など、各登録局（RA）の要求する暗号方式に従ってモジュールを選択して署名を実行し、各方式による署名を行なった公開鍵証明書の発行を実行する。

【0099】

先に図3を用いて説明したように、公開鍵証明書の発行要求は、認証局（CA）のCAサーバ71によって受付られ、CAサーバの有する各登録局（RA）81～85と適用する署名アルゴリズムの種類を対応付けたテーブルに従って、署名モジュール72a～72nが選択され、選択された署名モジュールに生成した公開鍵証明書を送信するとともに署名実行命令を送信する。公開鍵証明書と署名実行命令を受信した各署名モジュールは、それぞれのモジュールによって実行可能な署名アルゴリズム（ex. RSA, ECC）に従って署名処理を実行し、署名処理済みの公開鍵証明書をCAサーバ71に返送し、CAサーバ71は、署名済みの公開鍵証明書を発行要求を出した登録局（RA）81～85に送付する。

【0100】

認証局（CA）である電子認証装置を構成するCAサーバについて、図10を用いて説明する。認証局（CA）は、CAサーバ100と複数の署名アルゴリズムを実行する署名モジュール150を有し、署名モジュール（HSM）150に対してHSMインターフェース114を介して、先に説明した構成データ（図4, 5参照）を含む公開鍵証明書を送信し、署名モジュール150で署名を実行して、署名済みの公開鍵証明書をHSMインターフェース114を介して受信する。CAサーバ100と署名モジュール150との接続は、例えばPCIバス、あるいはイーサネット等のネットワーク接続される。いずれもセキュアな通信路として設定される。

【0101】

CAサーバ100は、認証局（CA）サーバの管理する登録局（RA）の情報を管理するRA管理データベース121、公開鍵証明書の検証用の鍵を格納した検証鍵データベース122、発行済みの公開鍵証明書、公開鍵証明書のリスト、公開鍵証明書の失効リスト情報を格納したリポジトリデータベース123を有し、データベースインターフェース（DB-I/F）116を介してこれら各データベースにアクセスする。

【0102】

各登録局（RA）181～183とは、相互認証処理を実行して通信相手の確認を実行した後、公開鍵証明書発行要求をネットワークインターフェース115を介して受信し、前述のデータ項目を持つ公開鍵証明書を生成して、RA管理データベース121に管理された情報、あるいは各登録局（RA）181～183からの要求に基づいて、署名アルゴリズムを決定する。その後、決定した署名アルゴリズムを実行する署名モジュール150-xを選択してHSMインターフェース114を介して生成した公開鍵証明書を送信する。この際、必要であれば、鍵長、パラメータなど署名生成に必要なデータも併せて送信する。公開鍵証明書を受信した署名モジュール150-xでは、モジュールの実行可能な例えばRSA、ECDSAなどの署名アルゴリズムに従って署名を実行して、CAサーバに返送する。CAサーバは、署名済みの公開鍵証明書を受信すると、検証鍵データベースから対応する検証鍵を取り出して検証し、署名が正しく行われていることを確認し、さらにリポジトリデータベース123に登録した後、署名済みの公開鍵証明書を発行要求元の登録局（RA）に送信する。

【0103】

CPU111は、上述の一連の処理を制御し、またRAM112、ROM113は、各処理に必要な処理プログラムの格納領域、ワークエリアの提供領域として機能する。さらに、表示部117、入力部118はオペレータによるデータ、コマンドの入力または表示処理に用いられる。

【0104】

RA管理データベース121のデータ構成例を図11に示す。RA管理データベースは、認証局（CA）が公開鍵証明書の発行処理の受付を行なう登録局（R

A) 毎に適用する署名アルゴリズムを対応付けたテーブルを格納する。

【0105】

図11に示す構成において、登録局（RA0001）は、署名方式としてRSA署名方式を用い、RSA署名に用いる鍵長は1024ビット（bit）である。このRSA署名を実行する署名モジュールはHSM-001であることが示されている。

【0106】

登録局（RA0002）は、署名方式としてRSA署名方式を用い、RSA署名に用いる鍵長は2048ビット（bit）である。このRSA署名は、署名モジュール：HSM-002, 003, 004の3つのモジュールを併用する負荷分散方式により実行されることを示している。

【0107】

登録局（RA0003）は、署名方式としてRSA署名方式とECC署名方式の複数署名方式を適用している。複数署名方式において【○】がチェックされている場合は、登録局が複数の署名方式を許容していることを示している。複数署名方式は、後段でさらに説明するが、例えばECC、RSA署名のいずれかを選択して実行することを登録局側から認証局（CA）に対して要求し、認証局が要求に従って署名アルゴリズムを選択する場合と、認証局が、1つの公開鍵証明書に複数の署名アルゴリズムによる複数の署名を付加する場合とがある。なお、署名方式が、ECDSA署名方式（Elliptic Curve Digital Signature Algorithm）である場合は、鍵長以外に、利用する楕円曲線パラメータがテーブルに格納される。

【0108】

署名を実行する署名モジュールに鍵長データやパラメータが固定的に設定されている場合は、署名モジュールに設定された鍵長、パラメータに基づいて署名が実行されるが、署名モジュールが複数の鍵長、またはパラメータに基づく署名を実行可能な構成である場合は、CAサーバから、RA管理データベースに登録された鍵長、または鍵長およびパラメータを署名モジュールに送信して署名モジュールがCAサーバから受信した鍵長、または鍵長およびパラメータに従って署名

を実行する。

【0109】

図12に検証鍵データベースの構成例を示す。検証鍵データベースは、署名処理を実行する署名モジュール識別子に対応して、その署名モジュールで実行される署名の検証用の鍵を格納したデータベースである。図12に示すように、署名モジュール識別子としてのHSM-ID、当該署名モジュールで実行される署名方式、使用される鍵長、パラメータ、および検証鍵が対応付けられて格納されている。

【0110】

ここで検証鍵は、各署名モジュールが署名鍵として生成した秘密鍵に対応する公開鍵であり、例えばCAサーバが各署名モジュールから受信して検証鍵データベースに格納する。CAサーバは、署名済みの公開鍵証明書を署名モジュールから受信すると、検証鍵データベース122から対応する検証鍵（公開鍵）を取り出して、署名検証を実行し署名の正しいことを確認して、証明書要求元の登録局（RA）に送信する。

【0111】

次に電子認証装置の署名モジュールを格納したハードウェア・セキュリティ・モジュール（HSM）の構成について図13を用いて説明する。HSMの各々は、固有の署名アルゴリズムを実行する署名モジュールを有する。HSMの各々は、耐タンパ構成を持ち、HSMの取り外しにより、署名生成用の秘密鍵情報などの格納情報が消去される。

【0112】

図13に示すように、HSM150は、CAサーバとのデータ送受信を実行する通信インターフェース152、HSM内の処理制御を行なうCPU151、RSA、ECDSAなど所定の署名アルゴリズムを実行する署名モジュール160、HSM識別子（ID）などの各種HSM固有データを格納した不揮発性メモリ153、署名アルゴリズムの識別処理、鍵長、パラメータ情報など、CAサーバからの署名要求データから取得するための解析処理を実行するプログラムなど、各種処理プログラムを格納したROM154、署名モジュールにおいて生成あるいは

は外部から受信した秘密鍵、署名アルゴリズム識別情報、鍵長、パラメータ情報など可変設定用情報を格納するRAM155を有する。

【0113】

署名モジュール160は、乱数発生ユニット161、署名生成ユニット162、ハッシュ計算ユニット163を有し、例えばECDSA署名アルゴリズムを実行する場合は前述の図6で説明した処理フローに従った署名処理を実行し、RSA署名アルゴリズムを実行する場合は前述の図9で説明した処理フローに従った署名処理を実行する。

【0114】

[署名モジュールに対する鍵格納処理]

署名モジュールの各々は前述したようにRSA、ECDSAなど特定の署名アルゴリズムに従った署名を実行する。この署名処理に使用する署名鍵となる秘密鍵は、個々の署名モジュール内で生成する構成としてもよいが、ある特定の署名モジュールで生成した鍵を他の署名モジュールに送信して格納する構成としてもよい。複数の署名モジュールに同一の鍵を格納する場合には、このように外部で生成した鍵を複数の署名モジュールに提供する構成が有効である。

【0115】

署名鍵は漏洩を防止するため、署名鍵読み込みや書き込みの操作を行う際には、専用ソフトウェアによるパスワード等を用いたオペレータの認証処理、および書き込みを行う署名モジュールの識別番号等のチェックなど、厳密な運用をすることが望ましい。

【0116】

図14～図17を用いて署名鍵のある特定の署名モジュールで生成して他の署名モジュールを持つHSMに格納する処理について説明する。図14は、署名モジュール1において署名鍵を生成し、複数の他の署名モジュール2～nに送信する処理を説明する図である。なお図中の専用ソフトは、例えばCAサーバのROMに格納されたプログラムである。オペレータはCAサーバと各署名モジュール間で鍵生成命令、生成鍵の送受信処理を実行し、複数の署名モジュール(HSM)に対して生成した鍵(秘密鍵)を送信する。この場合、CAサーバの検証鍵デ

ータベースに対しては、生成した秘密鍵に対応する検証鍵（公開鍵）を各HSM識別子に対応付けて格納する。なお、署名鍵の生成命令、鍵送受信時には、不正なデータ送受信を防止するため、パスワードなどによる認証処理をデータ送信側と受信側間で実行し、認証が成立したことを条件として生成した署名鍵他のデータ、コマンドを送信する。

【0117】

図15に例えばCAサーバにおいて実行される専用ソフトの処理フローを示す。当フローは、鍵生成命令を特定の署名モジュール（署名モジュール1）に対して実行し、署名モジュール1で生成した署名鍵を受信して他の署名モジュールに送信する処理を示している。

【0118】

まず、鍵生成処理、格納処理を実行するオペレータの認証を行なう（S101）。署名鍵の生成処理を実行可能なオペレータは予め登録された特定のオペレータのみであり、例えばパスワード、指紋認証などの処理によって正当なオペレータであることを確認（S102）する。正当なオペレータであることが確認されると、署名鍵の生成を依頼する署名モジュール1（HSM1）に対して鍵生成・送信命令を発行（S103）する。

【0119】

次に、署名モジュール1（HSM1）鍵生成・送信命令を発行した機器、例えばCAサーバ間で相互認証を実行する。相互認証は、共通鍵方式、あるいは公開鍵方式による相互認証が実行される。認証用の鍵は、それぞれの機器に予め格納されている。相互認証が成立（S105）すると、署名鍵、検証鍵が署名モジュールから受信されるまで待機し、署名鍵、検証鍵を署名モジュールから受信（S106）すると検証鍵を前述の検証鍵データベースに格納（S107）し、さらに同一の署名鍵を用いる他の署名モジュール（HSM）に署名鍵保存命令を出力（S108）し、相互認証を実行（S109）し、相互認証が成立したことを条件（S110）として、署名鍵を送信して、鍵の格納が済んだことを示すアクノレッジを署名モジュールから受信して鍵の書き込み処理終了を確認（S111）する。ステップS108-S111の処理を、署名鍵の書き込み対象署名モジュ

ール（HSM）に対して順次実行し、すべての署名モジュールに対する鍵書き込みが終了（S112）して処理を終了する。

【0120】

次に、図16を用いて署名鍵の生成処理を実行する署名モジュール1における処理を説明する。ステップS121において、鍵生成、送信命令を受信すると、専用ソフト（CAサーバ）との間で相互認証を実行（S122）し、相互認証が成立したことを条件（S123）として署名鍵、検証鍵の鍵ペアを生成（S124）し、正しく生成できたことを条件（S125）として、署名鍵を自デバイス内に保存（S126）する。さらに、生成した署名鍵と検証鍵を専用ソフト（CAサーバ）に送信（S127）し、送信成功を条件（S128）として処理を終了する。

【0121】

次に、図17を用いて署名鍵の格納処理を実行する署名モジュール2～Nにおける処理を説明する。ステップS131において、署名鍵保存命令を受信すると、専用ソフト（CAサーバ）との間で相互認証を実行（S132）し、相互認証が成立したことを条件（S133）として署名鍵の保存処理を実行（S134）し、正しく保存できたことを条件（S135）として、保存完了通知を専用ソフト（CAサーバ）に送信（S136）し、送信成功を条件（S137）として処理を終了する。

【0122】

[CAサーバにおける処理]

次に、CAサーバにおける処理を（1）署名モジュール（HSM）における鍵生成処理実行時、（2）公開鍵証明書発行および署名処理実行時に分けて説明する。

【0123】

まず、（1）署名モジュール（HSM）における鍵生成処理実行時の処理を図18を用いて説明する。左側がCAサーバ、右側が署名モジュールを有するHSMでの処理を示している。なお、図18におけるフローは認証処理等を省略しているが、図16、17と同様、CAサーバと署名モジュール間では、外部からの

一方、ステップS253において、負荷分散対象でない場合は、ステップS254に進み、複数署名方式対応であるかをRA管理データベース（図11参照）に基づいて判定する。複数署名方式である場合は、ステップS255において、公開鍵証明書発行要求中の要求署名方式に対応するデータエントリがRA管理データベースにあるか否かを判定し、ある場合（S256でYes）にのみ、ステップS257に進み、データベースからRA識別子（ID）に対応するHSM識別子（ID）を取得して取得したHSM識別子（ID）を持つHSMを処理モジュールとして決定する。

【0129】

なお、証明書発行要求に署名方式、鍵長、パラメータ指定がある場合、CAサーバは指定データに従って、指定の処理を実行可能なHSMをRA管理データベースから指定された署名方式、鍵長、パラメータによる検索を実行してHSMの選択処理を実行してもよい。

【0130】

このような手順で署名処理を実行すると、図19のステップS233において、決定したHSMに生成した公開鍵証明書データを送信する。公開鍵証明書データを受信（S241）したHSMは署名処理を実行（S242）し、署名済み証明書をCAサーバに送信（S243）する。

【0131】

HSMから署名済み証明書を受信（S234）したCAサーバは、検証鍵データベースからHSM識別子に基づいて検証鍵を取り出し（S235）、取り出した検証鍵を用いて署名検証処理を実行（S236）し、検証が成功（S237）した後、署名済み証明書を要求元の登録局（RA）に送信して処理を終了する。

【0132】

〔公開鍵証明書の発行処理例〕

次に、エンドエンティティ（EE）、登録局（RA）、認証局（CA）間において実行される公開鍵証明書の代表的な発行例を説明する。

【0133】

図21にエンドエンティティ（EE）300、登録局（RA）311, 312

、認証局（CA）サーバ321、署名モジュールを持つHSM331, 332, 333の構成例を示す。エンドエンティティ（EE）300は、登録局（RA）311, 312を介して公開鍵証明書発行要求を認証局（CA）サーバ321に対して行なう。

【0134】

このとき、認証局（CA）サーバのRA管理データベースには図21（a）に示すようなデータが登録され、また検証鍵データベースには図21（b）に示すデータが登録されているとする。図21（b）に示すRA管理データベースから明らかのように、登録局RA1は、署名方式RSA、鍵長1024ビットの署名を許容し、その署名を実行するモジュールはHSM1である。また、登録局RA2は、複数署名方式を許容し、署名方式RSA、鍵長2048ビットの署名と、署名方式ECDSA、鍵長192ビット、パラメータ $p = xx\dots$ 、さらに署名方式ECDSA、鍵長192ビット、パラメータ $p = yy\dots$ の各署名方式を許容し、登録局RA2のRSA署名を実行するモジュールはHSM2であり、ECDSA署名を実行するモジュールはHSM3である。また、検証鍵データベースには、各HSMの署名方式、鍵長、パラメータ情報と検証鍵が格納されている。

【0135】

このような設定の下、エンドエンティティ（EE）300からの公開鍵証明書発行依頼が各登録局（RA）に対して行われる。その各態様について順次説明する。

【0136】

まず、図22に示す態様は、エンドエンティティ（EE）300から登録局（RA1）311に対して公開鍵証明書発行依頼が出力された例を示す。図中の（1）～（10）は処理順を示している。以下、（1）～（10）の順に処理内容を説明する。

【0137】

（1）エンドエンティティ（EE）300はユーザデータなど公開鍵証明書発行に必要データを登録局（RA1）311に対して送信し、公開鍵証明書発行依頼を実行する。

(2) 登録局 (RA1) 311は、エンドエンティティ (EE) 300からの正当な証明書発行要求であることを確認し、ユーザ登録処理など、必要な処理を実行した後、

(3) 証明書発行要求を行なう。

証明書発行要求には、図22(a)に示すように、証明書発行要求コマンド、証明書格納データなど必要なメッセージデータ、さらに、登録局識別子 (ID) を含む。

【0138】

(4) 証明書発行要求を受信したCAサーバ321は、RA管理データベースを参照して署名を実行するHSMを決定する。この場合、図21(a)に示すRA管理データースからHSM1が署名実行モジュールとして選択される。

(5) CAサーバ321は、HSM1, 331に対して署名生成命令を出力する。

署名生成命令には、図22(b)に示すように、署名生成命令コマンドと、生成した証明書のメッセージデータを含む。また、HSM1が可変長の鍵を生成可能なHSMであれば鍵長を指定するデータを含む場合がある。

【0139】

(6) HSM1, 331は、署名生成命令に従って、署名処理を実行する。この場合は、RSA方式の署名を実行する。

(7) HSM1, 331は署名終了後、署名済み公開鍵証明書をCAサーバ321に送信する。

(8) CAサーバ321は、検証鍵データベースから検証鍵を取り出して受信した公開鍵証明書の署名の検証処理を実行する。

(9) 検証が成立すると、署名済み公開鍵証明書を要求元の登録局 (RA1) 311に送信し、

(10) 登録局 (RA1) 311は受信した署名済み公開鍵証明書を要求元のエンドエンティティ (EE) 300に送信する。

【0140】

次に、図23に示す態様は、エンドエンティティ (EE) 300から登録局 (

R A 2) 3 1 2 に対して公開鍵証明書発行依頼が出力された例を示す。図中の(1)～(10)は処理順を示している。以下、(1)～(10)の順に処理内容を説明する。

【0141】

(1) エンドエンティティ (E E) 3 0 0 はユーザデータなど公開鍵証明書発行に必要データを登録局 (R A 2) 3 1 2 に対して送信し、公開鍵証明書発行依頼を実行する。

(2) 登録局 (R A 2) 3 1 2 は、エンドエンティティ (E E) 3 0 0 からの正当な証明書発行要求であることを確認し、ユーザ登録処理など、必要な処理を実行した後、

(3) 証明書発行要求を行なう。

証明書発行要求には、図23(a)に示すように、証明書発行要求コマンド、証明書格納データなど必要なメッセージデータ、登録局識別子 (I D) 、さらに署名方式、鍵長、パラメータの指定データを含む。

【0142】

(4) 証明書発行要求を受信したCAサーバ321は、RA管理データベースを参照して署名を実行するHSMを決定する。この場合、図21(a)に示すRA管理データースからHSM3が署名実行モジュールとして選択される。

(5) CAサーバ321は、HSM3, 333に対して署名生成命令を出力する。

署名生成命令には、図23(b)に示すように、署名生成命令コマンドと、生成した証明書のメッセージデータ、鍵長、パラメータを指定するデータを含む。

【0143】

(6) HSM3, 333は、署名生成命令に従って、署名処理を実行する。この場合は、ECDSA方式の署名を実行する。

(7) HSM3, 333は署名終了後、署名済み公開鍵証明書をCAサーバ321に送信する。

(8) CAサーバ321は、検証鍵データベースから検証鍵を取り出して受信した公開鍵証明書の署名の検証処理を実行する。

(9) 検証が成立すると、署名済み公開鍵証明書を要求元の登録局（RA2）312に送信し、

(10) 登録局（RA2）312は受信した署名済み公開鍵証明書を要求元のエンドエンティティ（EE）300に送信する。

【0144】

次に、図24に示す態様は、エンドエンティティ（EE）300から登録局（RA2）312に対して公開鍵証明書発行依頼が出力され、同時に複数の署名を実行する例を示す。図中の（1）～（14）は処理順を示している。以下、（1）～（14）の順に処理内容を説明する。

【0145】

(1) エンドエンティティ（EE）300はユーザデータなど公開鍵証明書発行に必要データを登録局（RA2）312に対して送信し、公開鍵証明書発行依頼を実行する。

(2) 登録局（RA2）312は、エンドエンティティ（EE）300からの正当な証明書発行要求であることを確認し、ユーザ登録処理など、必要な処理を実行した後、

(3) 証明書発行要求を行なう。

証明書発行要求には、図24（a）に示すように、証明書発行要求コマンド、証明書格納データなど必要なメッセージデータ、登録局識別子（ID）、さらに複数の署名方式、鍵長、パラメータの指定データを含む。

【0146】

(4) 証明書発行要求を受信したCAサーバ321は、RA管理データベースを参照して署名を実行するHSMを決定する。この場合、図21（a）に示すRA管理データベースからHSM2と、HSM3が署名実行モジュールとして選択される。

(5) CAサーバ321は、まず、HSM2、332に対して署名生成命令を出力する。

HSM2、332に対する署名生成命令には、図24（b）に示すように、署名生成命令コマンドと、生成した証明書のメッセージデータ、鍵長を指定するデ

ータを含む。

【0147】

(6) HSM2, 332は、署名生成命令に従って、署名処理を実行する。この場合は、ECDSA方式の署名を実行する。

(7) HSM2, 332は署名終了後、署名済み公開鍵証明書をCAサーバ321に送信する。

(8) CAサーバ321は、検証鍵データベースから検証鍵を取り出して受信した公開鍵証明書の署名の検証処理を実行する。

【0148】

(9) CAサーバ321は、次に、HSM3, 333に対して署名生成命令を出力する。

署名生成命令には、図24(c)に示すように、署名生成命令コマンドと、証明書のメッセージデータ、鍵長、パラメータを指定するデータを含む。

【0149】

(10) HSM3, 333は、署名生成命令に従って、署名処理を実行する。この場合は、ECDSA方式の署名を実行する。

(11) HSM3, 333は署名終了後、署名済み公開鍵証明書をCAサーバ321に送信する。

(12) CAサーバ321は、検証鍵データベースから検証鍵を取り出して受信した公開鍵証明書の署名の検証処理を実行する。

(13) 検証が成立すると、署名済み公開鍵証明書を要求元の登録局(RA2)312に送信し、

(14) 登録局(RA2)312は受信した署名済み公開鍵証明書を要求元のエンドエンティティ(EE)300に送信する。

【0150】

【署名モジュール構成の秘密管理構成】

次に、認証局としての電子認証装置の署名モジュールの構成を外部に対して秘密に管理する実施例について説明する。

【0151】

図25に本実施例の構成を説明するブロック図を示す。図25に示すように認証局(CA)401、登録局(RA)402の各々は、署名モジュール構成管理テーブルを有する。

【0152】

署名モジュール構成管理テーブルは、署名方式識別子(1～n)と、各署名方式を実行するHSMを対応付けたテーブルである。署名モジュール構成管理テーブルは、認証局(CA)401が各署名方式を実行するHSMに対して署名方式識別子(1～n)をランダムに対応付けて生成し、必要に応じてあるいは定期的に対応を更新する。生成されたテーブルは各登録局(RA)402に配布され、登録局(RA)402は、エンドエンティティからの証明書発行依頼に基づいて、証明書発行要求を署名方式識別子(1～n)を付加して生成し、認証局(CA)401に対して送信する。

【0153】

認証局(CA)401は、証明書発行要求中に含まれる署名方式識別子(1～n)に基づいて署名モジュール構成管理テーブルを検索し、HSMを選択して、選択されたHSMに対して署名要求を出力する。

【0154】

本方式によれば、署名モジュール構成管理テーブルが外部に漏洩しても署名方式が漏れることがないため、第三者による不正な署名処理などの発生が効果的に防止可能となる。なお、署名モジュール構成管理テーブルには認証局の署名を実行し、改竄防止を図ることが望ましい。

【0155】

【署名処理の分散方式】

次に署名処理を複数の署名モジュール(HSM)を利用して実行する構成について図26を用いて説明する。

【0156】

CAサーバ501は、登録局(RA)からの公開鍵証明書発行要求に従って、署名処理を実行する際、署名をいくつかのステップ(べき乗演算、ハッシュ値の生成等)に分割し、それぞれのステップを異なるHSMに処理要求を行なって、

各ステップを異なるHSM 601～60nが分担して処理を行う。

【0157】

例えばECC署名アルゴリズムの場合、図6を用いて説明したように、ハッシュ値生成処理、乱数生成処理、座標値Vの生成処理などECC署名の生成処理を複数ステップに分割することができる。これらの各処理を別モジュールにおいてそれぞれ個別に実行し、各モジュールの処理結果を異なるモジュールに転送し、次の処理を実行する。この場合、モジュール間での転送データには、データ送信側のモジュールが中間署名を実行し、受信モジュールは中間署名を含むメッセージデータに対してさらに中間署名を順次付加する。最終署名結果は、モジュール60nからCAサーバ501が受領する。署名生成には、すべてのモジュールが決められた順序にしたがって一連の処理を実行することが必要となる。

【0158】

また、単一のメッセージに対して複数の異なる署名づけを、各署名モジュール(HSM)で連続して実行する構成としてもよい。例えばHSM1, 601でRSA署名を実行し、HSM2, 602でECDSA署名aを実行し、HSMx, 60xで異なるパラメータによるECDSA署名xを実行する。これら複数の署名を行なった結果をCAサーバ501に送信する。この場合も、モジュール間での転送データには、データ送信側のモジュールが中間署名を実行し、受信モジュールは中間署名を含むメッセージデータに対してさらに中間署名を順次付加する。ただしこの場合は、CAサーバ501は、各モジュールでの検証鍵を個別に管理することが必要である。

【0159】

本構成によれば、認証局(CA)の管理する複数の署名モジュールの各々が一定の順序で公開鍵証明書のメッセージデータを転送することによって署名が生成されることになり、1つの署名モジュールのみを用いた署名生成は不可能になり、モジュールの漏洩による不正署名が防止可能となる。

【0160】

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が実施例の修正や代用を成し得る

ことは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参照すべきである。

【0161】

【発明の効果】

上述したように、本発明の公開鍵証明書発行システム、公開鍵証明書発行方法、および電子認証装置、並びにプログラム記憶媒体によれば、RSA、ECDSAなど様々な署名方式を実行する複数の署名モジュールを持ち、複数の署名方式を選択して実行可能な認証局(CA)を構成し、登録局(RA)に対応する署名方式に従って署名方式を選択して実行する構成とした。従って、特定の署名方式を実行する認証局を署名方式に応じて複数構成する必要がなく、1つの認証局のみで、様々な署名方式を許容する複数の登録局(RA)の要求に応じた各種の署名が可能となり、1つの認証局のみで複数の署名方式を持つ公開鍵証明書を発行することが可能となる。

【0162】

さらに、本発明の公開鍵証明書発行システム、公開鍵証明書発行方法、および電子認証装置、並びにプログラム記憶媒体によれば、RSA、ECDSAなど様々な署名方式を実行する複数の署名モジュールを持ち、複数の署名方式を選択して実行可能な認証局(CA)を構成したので、1つの公開鍵証明書に複数の異なる方式による署名を付加する処理が可能となる。

【0163】

さらに、本発明の公開鍵証明書発行システム、公開鍵証明書発行方法、および電子認証装置、並びにプログラム記憶媒体において、RSA、ECDSAなど様々な署名方式の一部ステップを実行する複数の署名モジュールを持ち、複数のモジュールを連携させて処理を実行して署名処理を実行する構成が可能となり、個別モジュールの漏洩による不正署名が防止可能となる。

【図面の簡単な説明】

【図1】

一般的な公開鍵証明書の例を示す図である。

【図2】

従来の公開鍵証明書の発行システムの概要を説明する図である。

【図3】

本発明の公開鍵証明書発行システムの概要を説明する図である。

【図4】

公開鍵証明書のデータ構成の詳細を説明する図（その1）である。

【図5】

公開鍵証明書のデータ構成の詳細を説明する図（その2）である。

【図6】

E C D S A署名生成処理の手順を説明するフロー図である。

【図7】

E C D S A署名検証処理の手順を説明するフロー図である。

【図8】

R S A署名処理に必要な鍵の生成手順を説明するフロー図である。

【図9】

R S A署名生成処理、検証処理の手順を説明するフロー図である。

【図10】

認証局（C A）サーバの構成を説明するブロック図である。

【図11】

認証局（C A）サーバの有する登録局（R A）管理データベース構成を説明する図である。

【図12】

認証局（C A）サーバの有する検証鍵データベース構成を説明する図である。

【図13】

署名モジュールを有するH S M構成を示すブロック図である。

【図14】

同一署名鍵を複数の署名モジュールで共有する方式を説明する図である。

【図15】

同一署名鍵を複数の署名モジュールで共有する処理を説明するフロー図（その

1) である。

【図16】

同一署名鍵を複数の署名モジュールで共有する処理を説明するフロー図（その2）である。

【図17】

同一署名鍵を複数の署名モジュールで共有する処理を説明するフロー図（その3）である。

【図18】

署名鍵を署名モジュールに格納する処理を説明するフロー図である。

【図19】

署名鍵を用いた署名モジュールにおける署名処理を説明するフロー図である。

【図20】

署名モジュールにおける署名処理実行時の署名モジュール決定処理を説明するフロー図である。

【図21】

署名モジュールにおける署名処理の具体例（例1）を説明する図である。

【図22】

署名モジュールにおける署名処理の具体例（例2）を説明する図である。

【図23】

署名モジュールにおける署名処理の具体例（例3）を説明する図である。

【図24】

署名モジュールにおける署名処理の具体例（例4）を説明する図である。

【図25】

署名モジュールの構成を秘密に管理する構成を説明する図である。

【図26】

複数の署名モジュールを連携して署名を実行する処理を説明する図である。

【符号の説明】

21 E C D S A認証局

22 E C D S A登録局

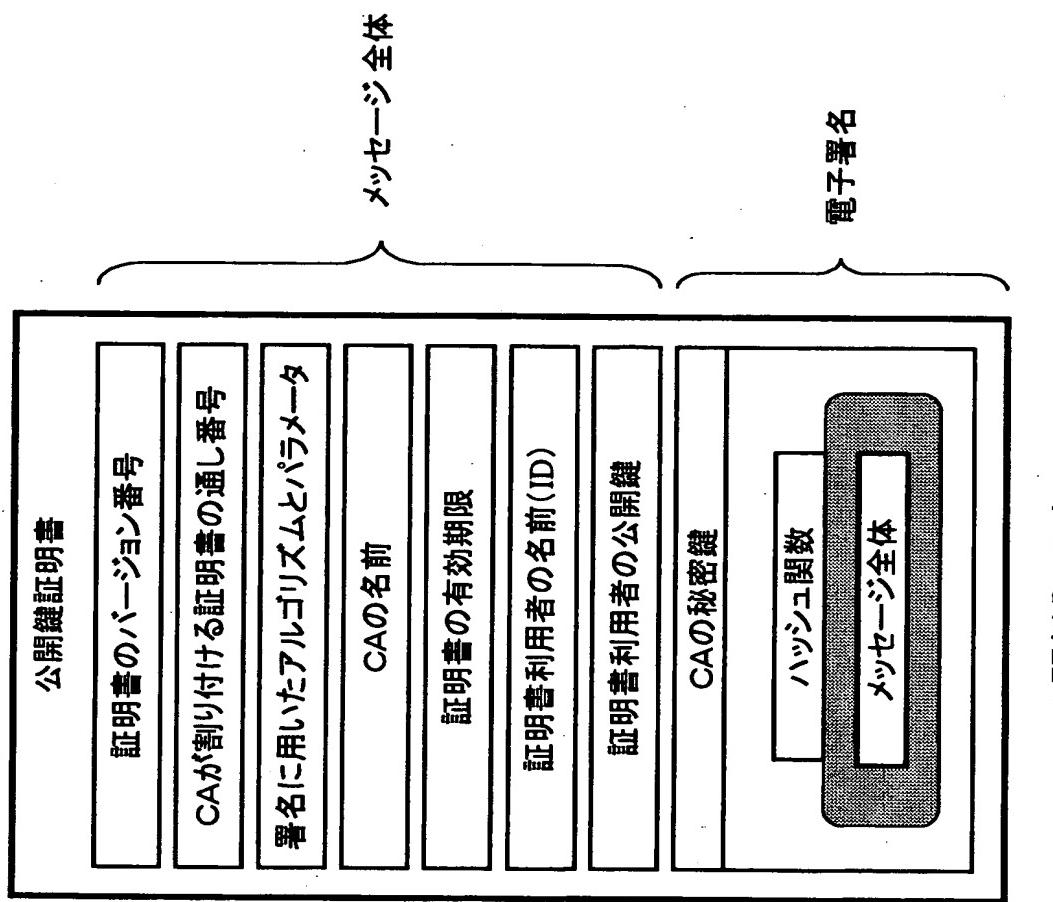
23 E C D S A デバイス
31 R S A 認証局
32 R S A 登録局
33 R S A デバイス
70 認証局 (C A)
71 C A サーバ
72 a ~ n 署名モジュール
81 ~ 85 登録局 (R A)
100 C A サーバ
111 C P U
112 R A M
113 R O M
114 H S M インタフェース
115 ネットワークインターフェース
116 データベースインターフェース
117 表示部
118 入力部
121 R A 管理データベース
122 検証鍵データベース
123 リポジトリ
150 署名モジュール
181 ~ 183 登録局 (R A)
151 C P U
152 通信インターフェース
153 不揮発性メモリ
154 R O M
155 R A M
160 署名モジュール
161 乱数発生ユニット

162 署名生成ユニット
163 ハッシュ計算ユニット
300 エンドエンティティ (EE)
311, 312 登録局 (RA)
321 CAサーバ
331~333 HSM
401 認証局サーバ
402 登録局
501 認証局 (CA) サーバ
601~60n HSM

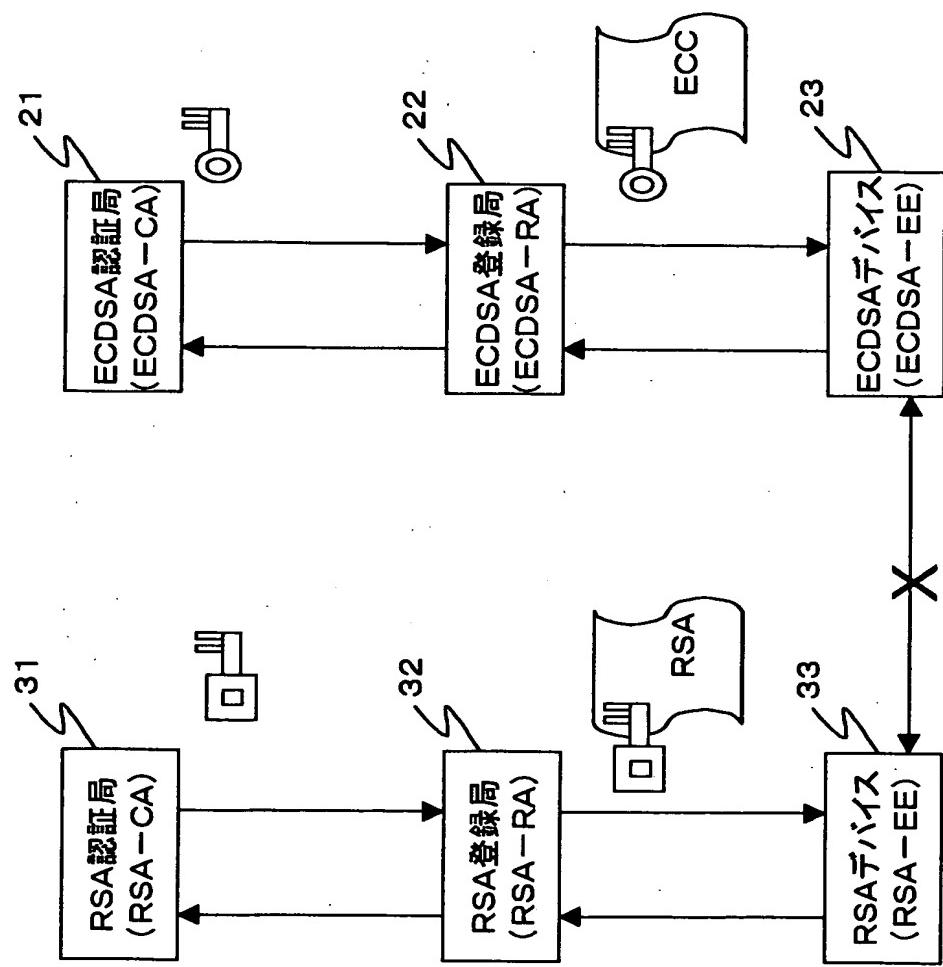
【書類名】

図面

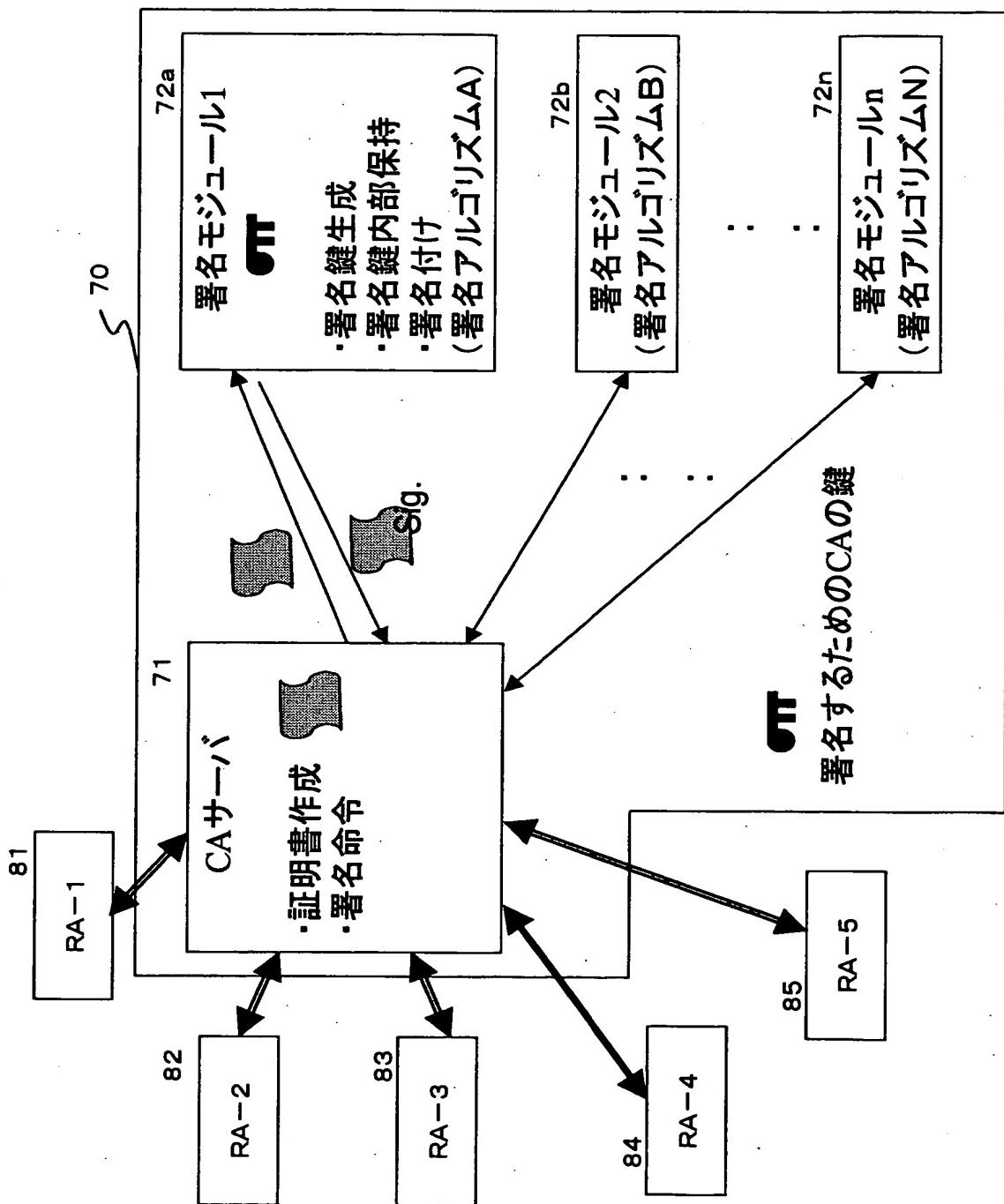
【図1】



【図2】



【図3】



【図4】

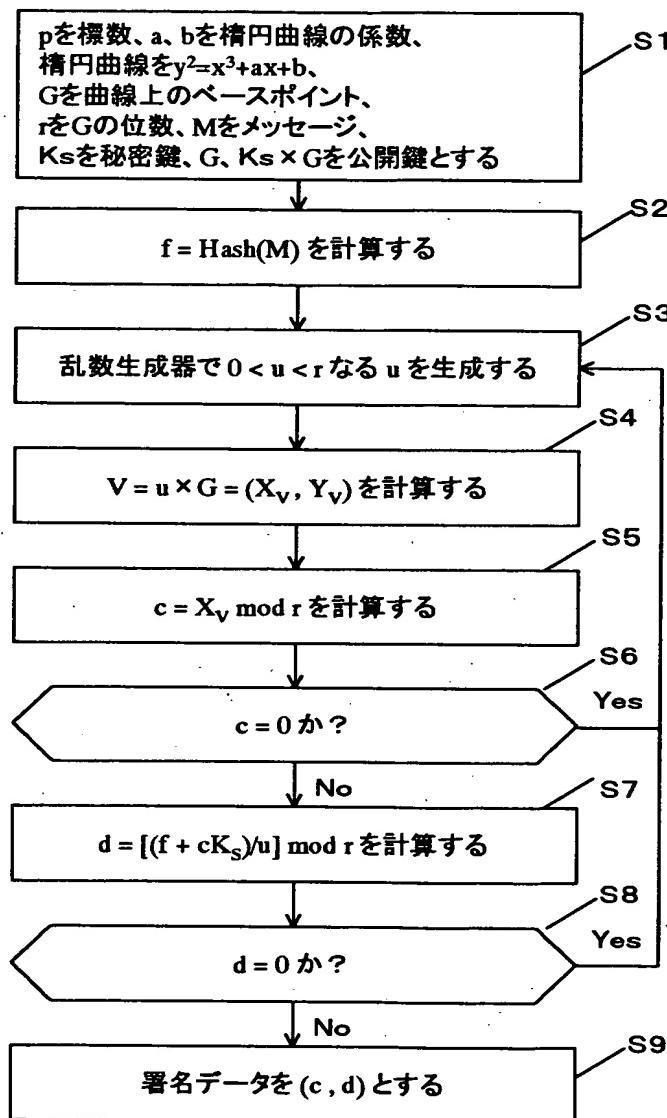
証明書フォーマット例 (X.509 V3に準拠)		
項目	説明	本IAにおける設定
Version1		
version	証明書のフォーマットのバージョン	V3
serial Number	IAによってつけられる証明書のSerial No.	シーケンシャルなシリアルナンバー
signature.algorith Identifier algorithm parameters	証明書の署名アルゴリズム、及びそのパラメータ	楕円曲線暗号/RSA 楕円の場合パラメータ RSAの場合鍵長
issuer	IA名(Distinguished Name形式)	本IAの名称
validity notBefore notAfter	証明書の有効期限 開始日時 終了日時	
subject	userを識別する名前	ユーザ機器IDまたはサービス主体のID
subject Public Key Info algorithm subject Public key	userの公開鍵情報 鍵のアルゴリズム 鍵	楕円曲線/RSA userの公開鍵
Version3		
authority Key Identifier key Identifier authority Cert Issuer authority Cert Serial Number	IAの署名確認用の鍵識別 鍵識別番号(8進数) IA名(General Name形式) 認証番号	
subject key Identifier	複数の鍵の証明をする場合	利用しない
key usage (0)digital Signature (1)non Repudiation (2)key Encipherment (3)data Encipherment (4)key Agreement (5)key CertSign (6)cRL Sign	鍵の使用目的を指定 (0)デジタル署名用 (1)否認防止用 (2)鍵の暗号化用 (3)メッセージの暗号化用 (4)共通鍵配達用 (5)認証の署名確認用 (6)失効リストの署名確認用	0,1,4,6を利用
private Key Usage Period notBefore notAfter	userに格納されている秘密鍵の有効期限。	証明書の有効期限=公開鍵の有効期限=秘密鍵の有効期限(default)

【図5】

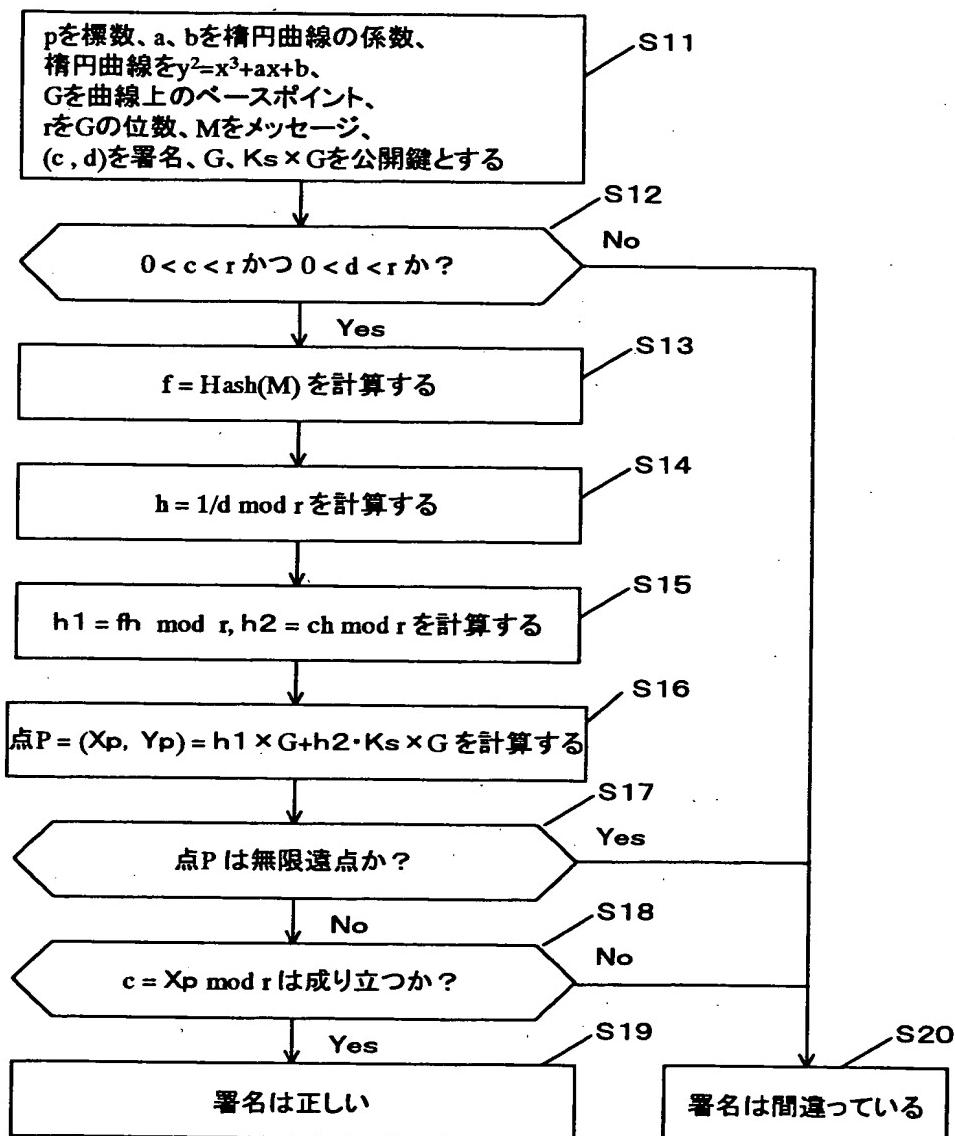
policy Mappings issuer Domain Policy subject Domain Policy	CA を認証する場合にのみ必要。発行認証局のポリシーと被認証ポリシーのマッピングを規定	default = なし
supported Algorithms algorithm Identifier intended Usage intended Certificate Policies	ディレクトリ(X.500)のアトリビュートを定義。コミュニケーションの相手がディレクトリ情報を利用する場合に、事前にそのアトリビュートを知らせるのに用いる。	default = なし
subject Alt Name	user の別名(GN 形式)。	利用しない
issuer Alt Name	項目は入れておく(default = なし)	default = なし
subject Directory Attributes	user の任意の属性。	利用しない
basic Constraints cA path Len Constraint	証明対象の公開鍵が認証局の署名用か、user のものかを区別	default = user用
name Constraints permitted Subtrees base minimum maximum excluded Subtrees	被認証者が認証局である場合(CA 認証)にのみ使用。	default = なし
policy Constraints requireExplicitPolicy inhibitPolicyMapping	認証パスの残りに対する明確な認証ポリシー ID、禁止ポリシーマップを要求する制限を記述	
CRL Distribution Points	user が証明書を利用する際に、証明書が失効していないかどうかを確認するための失効リストの参照ポイントを記述。	証明書を登録したところへのポインタ。失効リストは、発行元で管理
署名	発行者の署名	

【図6】

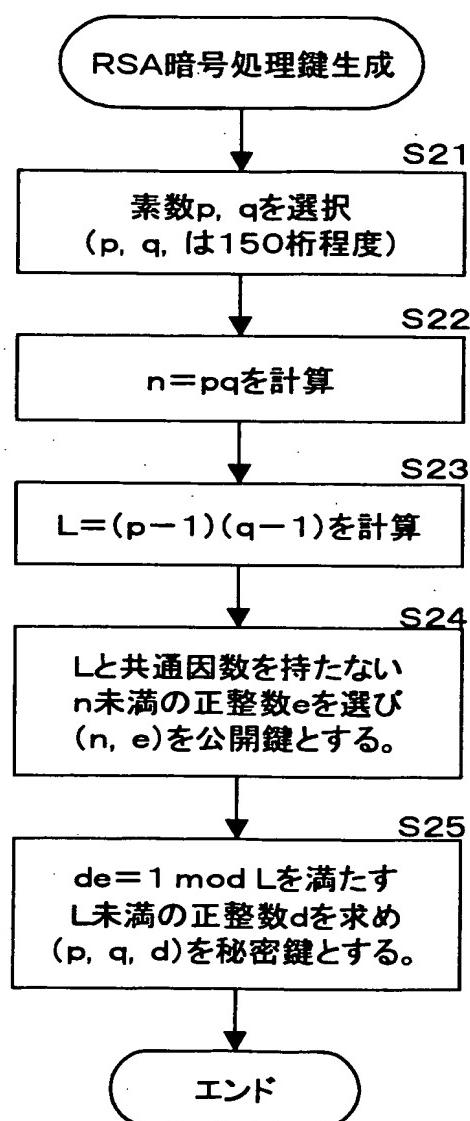
署名生成

署名生成(IEEE P1363/D3)

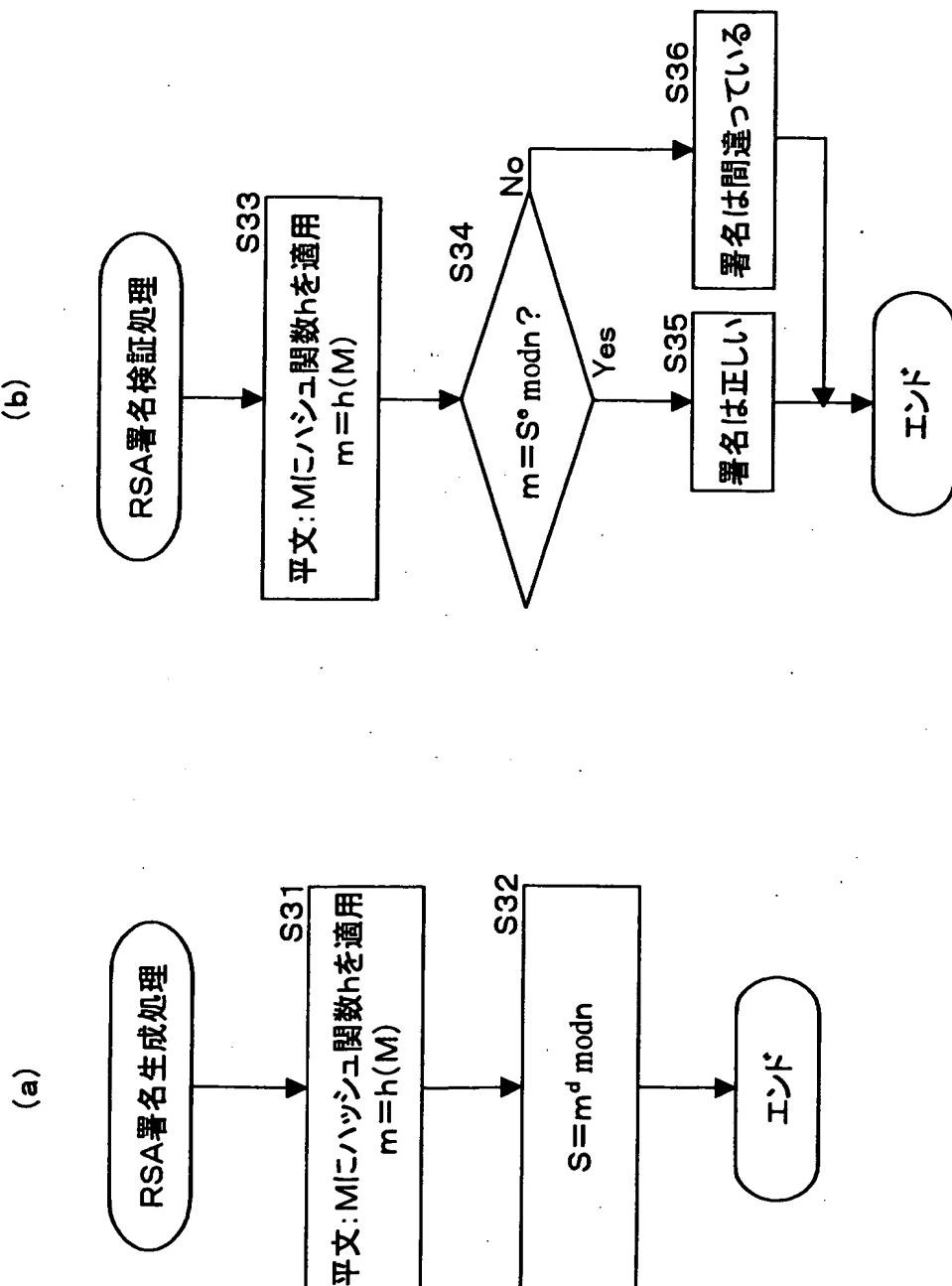
【図7】

署名検証署名検証(IEEE P1363/D3)

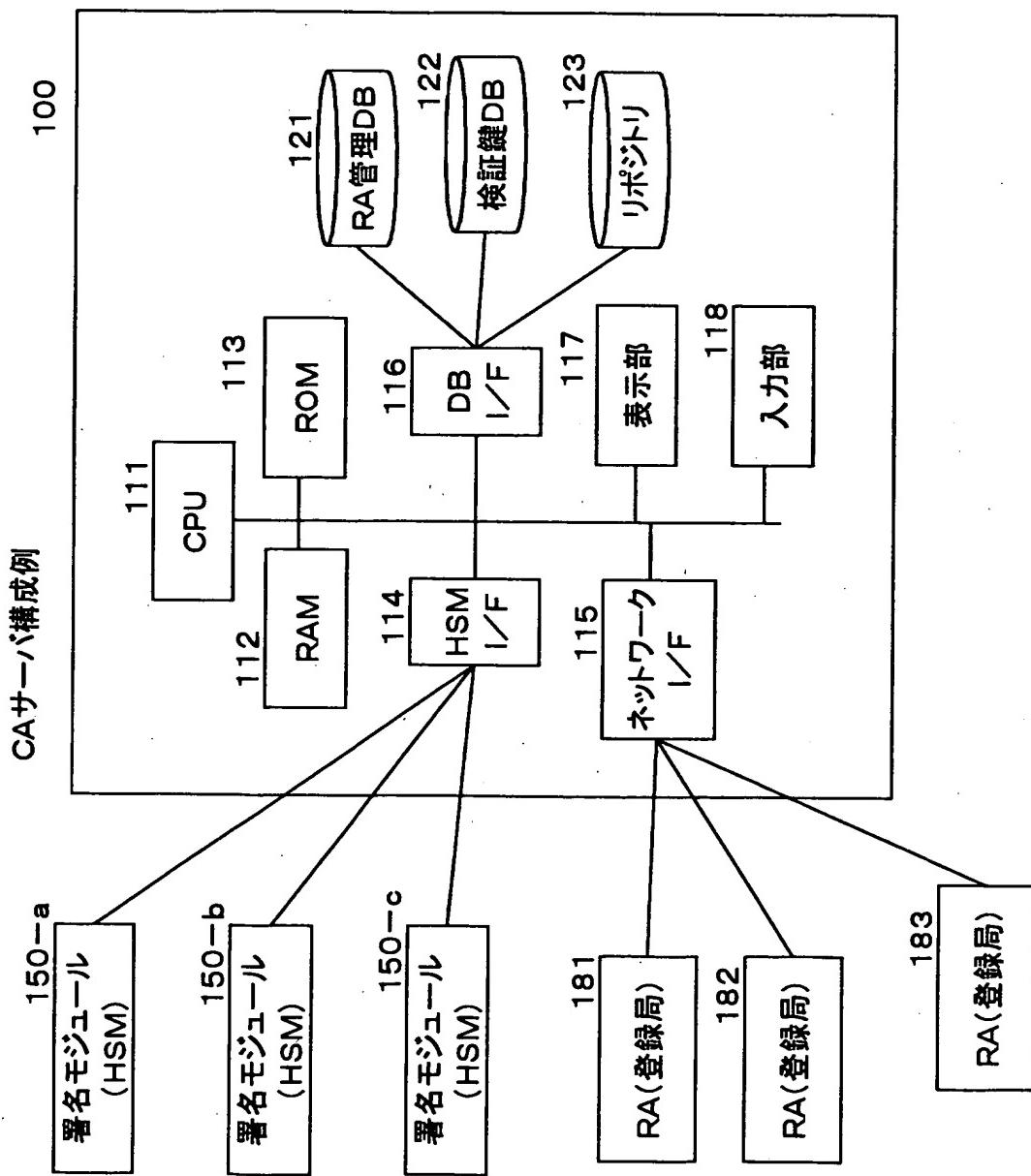
【図8】



【図9】



【図10】



【図11】

CAでのRA管理データベース例

RA ID	複数署名方式利用	署名方式	鍵長	パラメータ	負荷分散	利用HSM
RA0001	×	RSA	1024 bit	—	×	001
RA0002	×	RSA	2048 bit	—	○	002,003,004
RA0003	○	RSA	512 bit	—	×	005
RA0003	○	ECDSA	160 bit	$p=XX, \dots$	×	101
RA0004	○	RSA	1024 bit	—	×	006
RA0004	○	RSA	2048 bit	—	×	007
RA0004	○	ECDSA	192 bit	$p=YY, \dots$	×	102
RA0004	○	ECDSA	224 bit	$p=ZZ, \dots$	×	103

【図12】

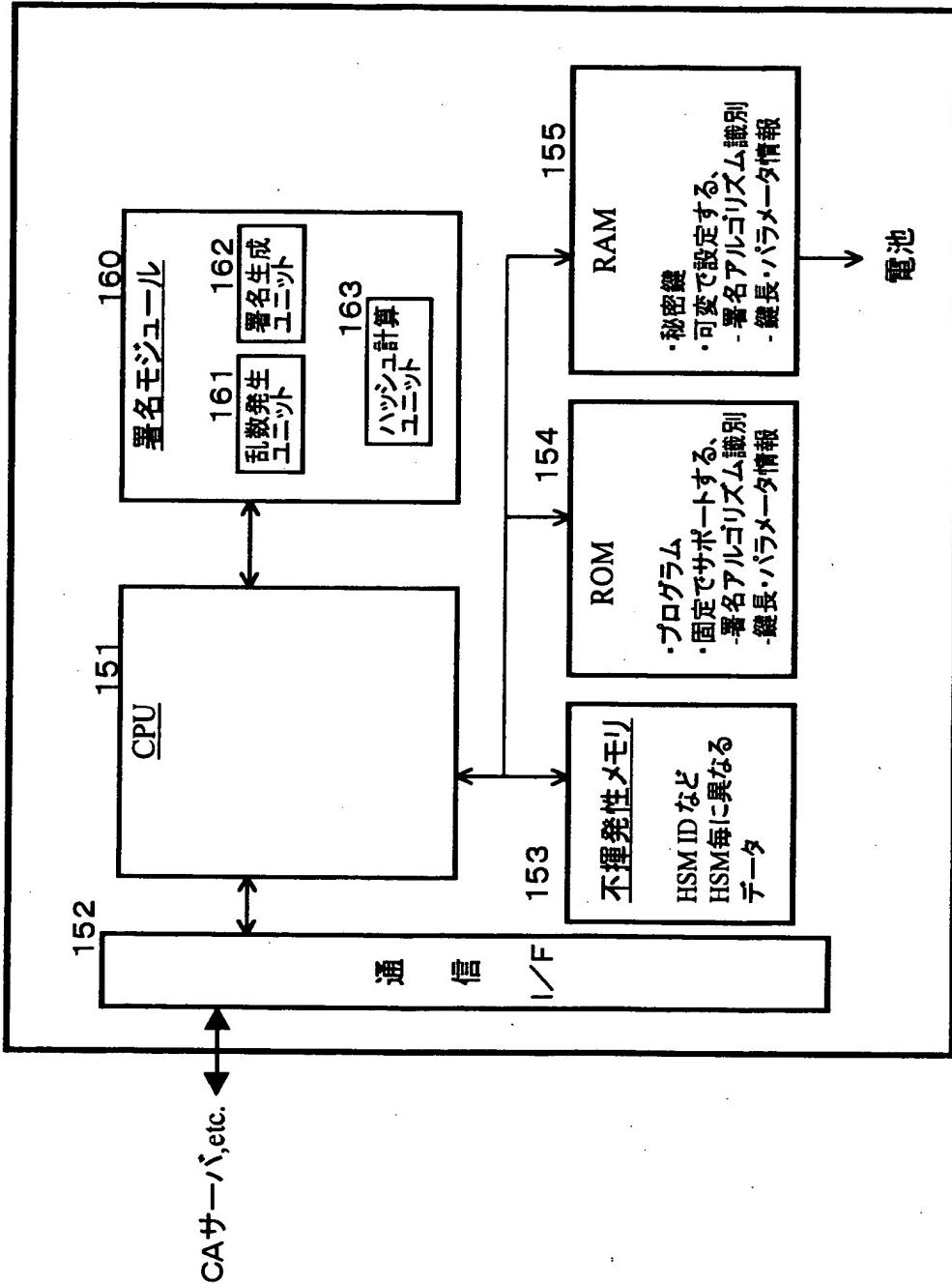
検証鍵データベース

HSM ID	署名方式	鍵長	パラメータ	検証鍵
201	RSA	2048 bit	—	◇□
"	RSA	1024 bit	—	◆□
202	ECDSA	160 bit	$p=XX, \dots$	△□
"	ECDSA	192 bit	$p=YY, \dots$	◀□

【図13】

150

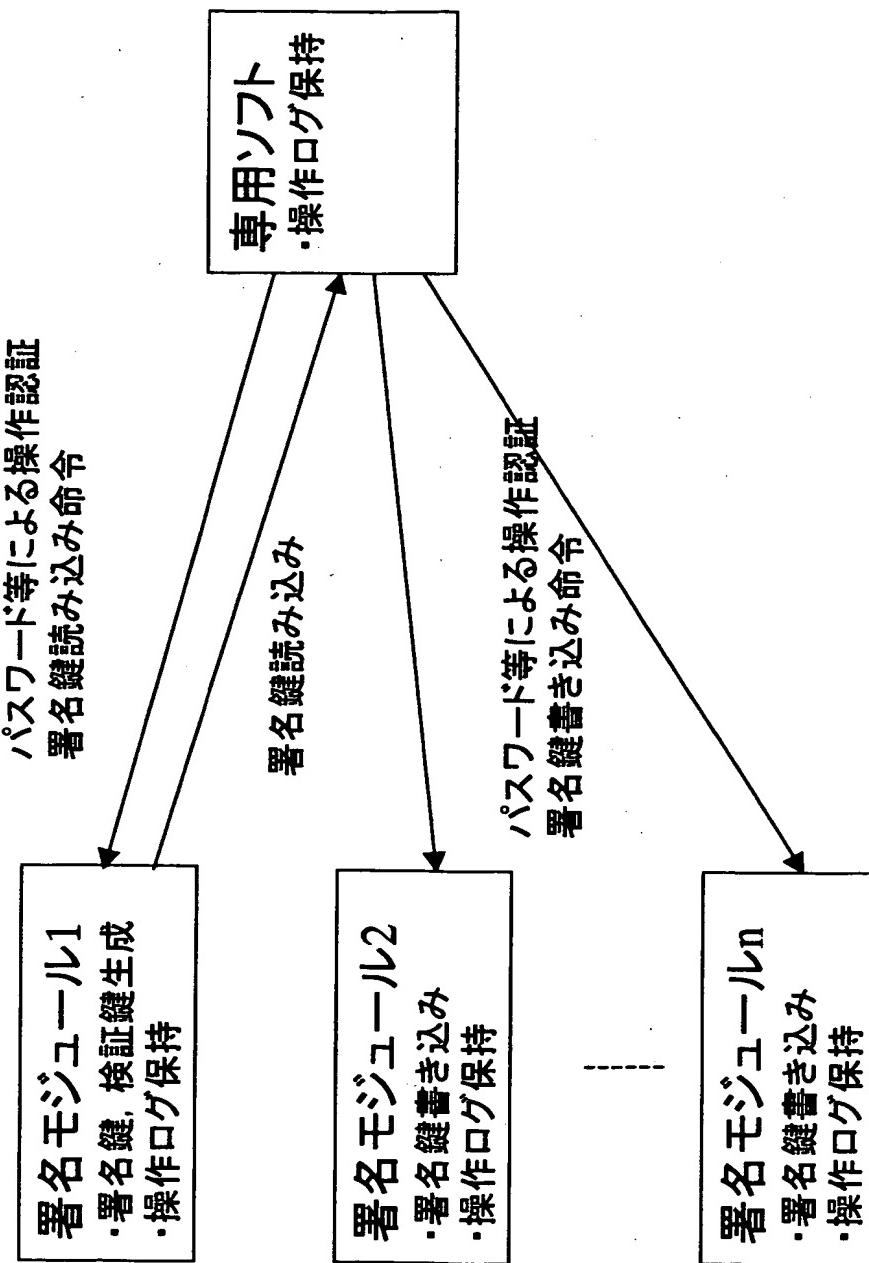
HSM構成例



※ HSM(例:ボード) 全体が耐タンパー性を持ち、取り外した場合には秘密鍵情報が消去されるように制御される

【図14】

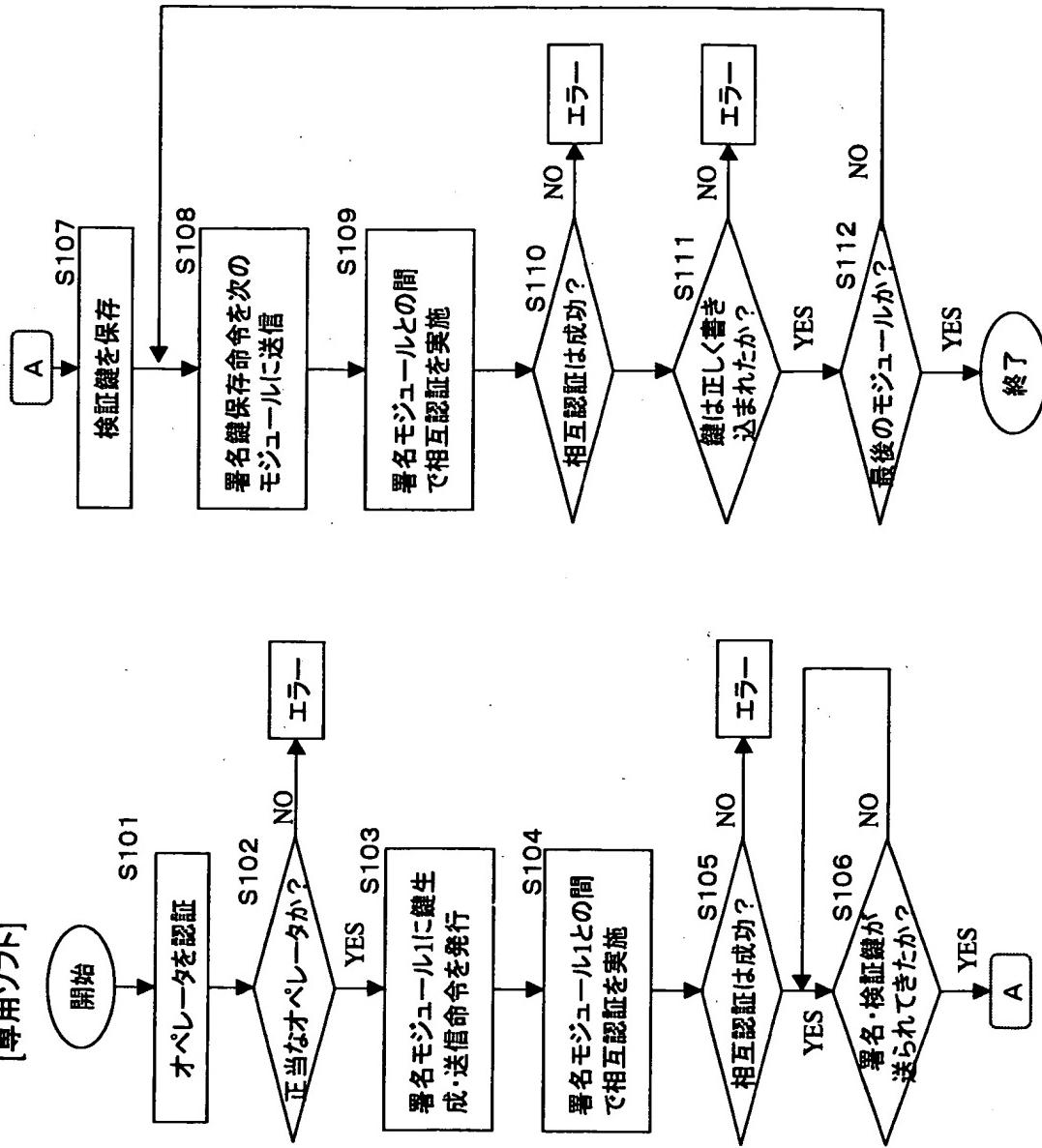
同一署名鍵を複数署名モジュールで共有する方式



【図15】

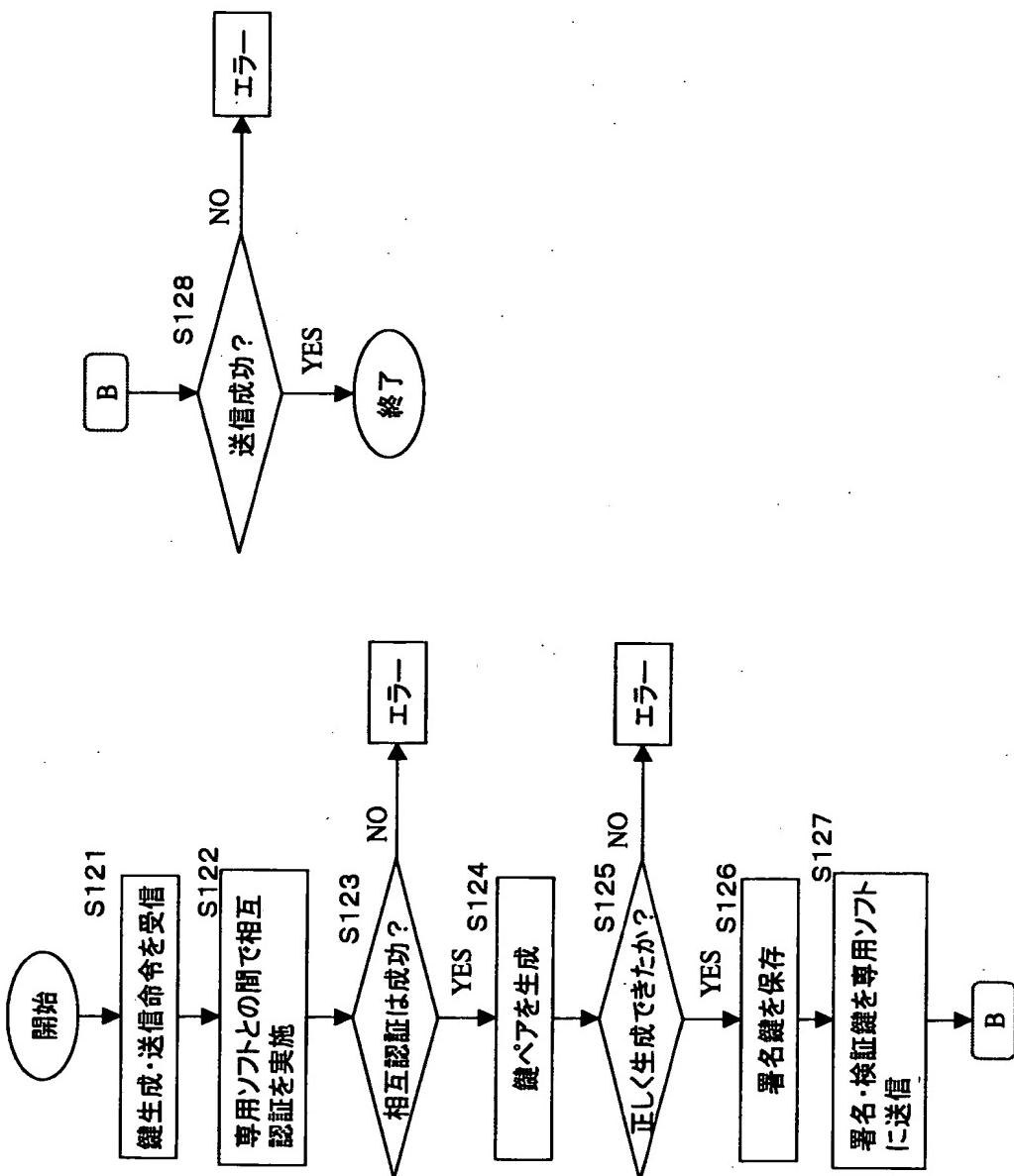
<鍵の共有> 同一署名鍵を複数署名モジュールで共有する方式

[専用ソフト]



【図16】

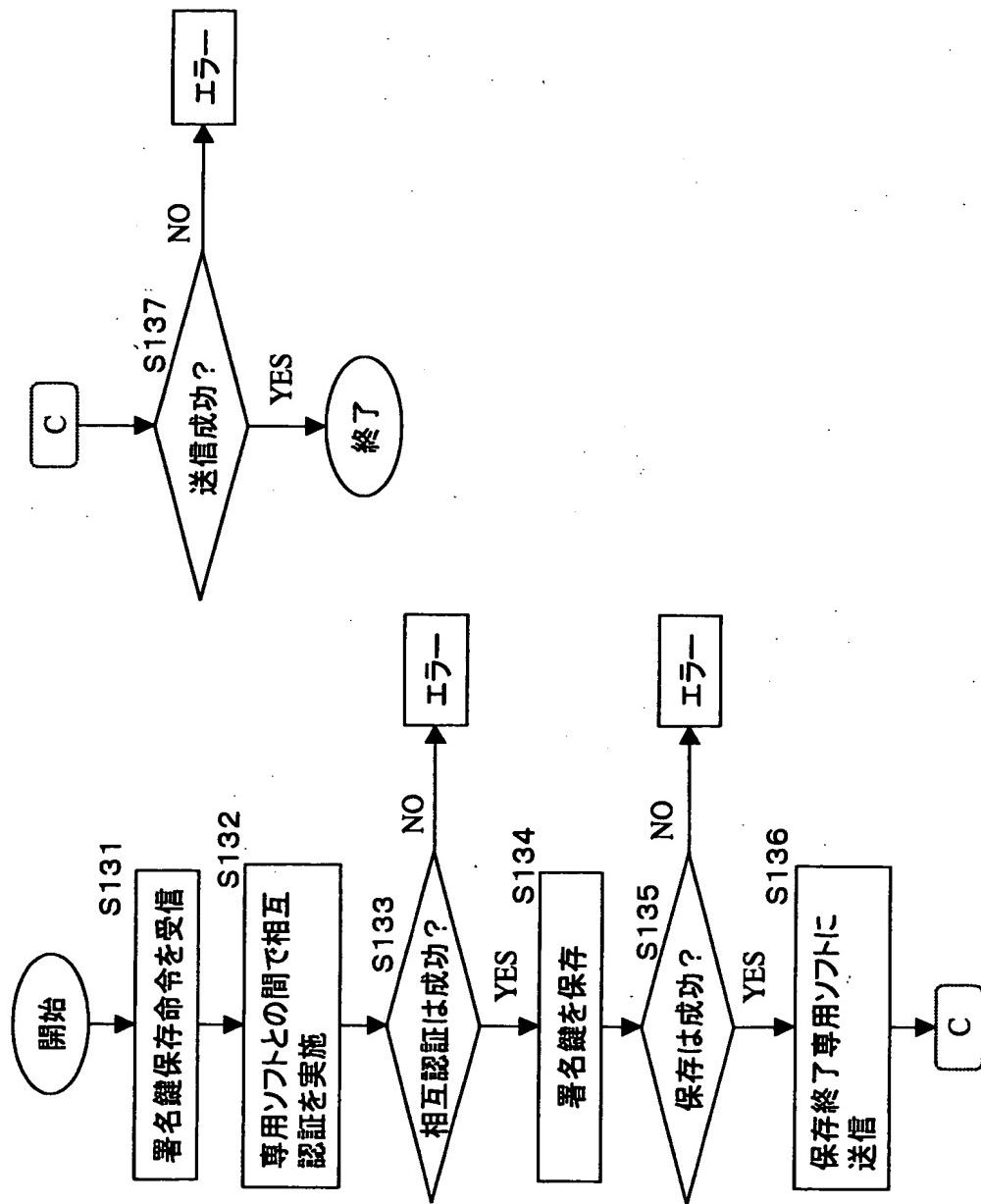
同一署名モジュールで共有する方式
[署名モジュール1]



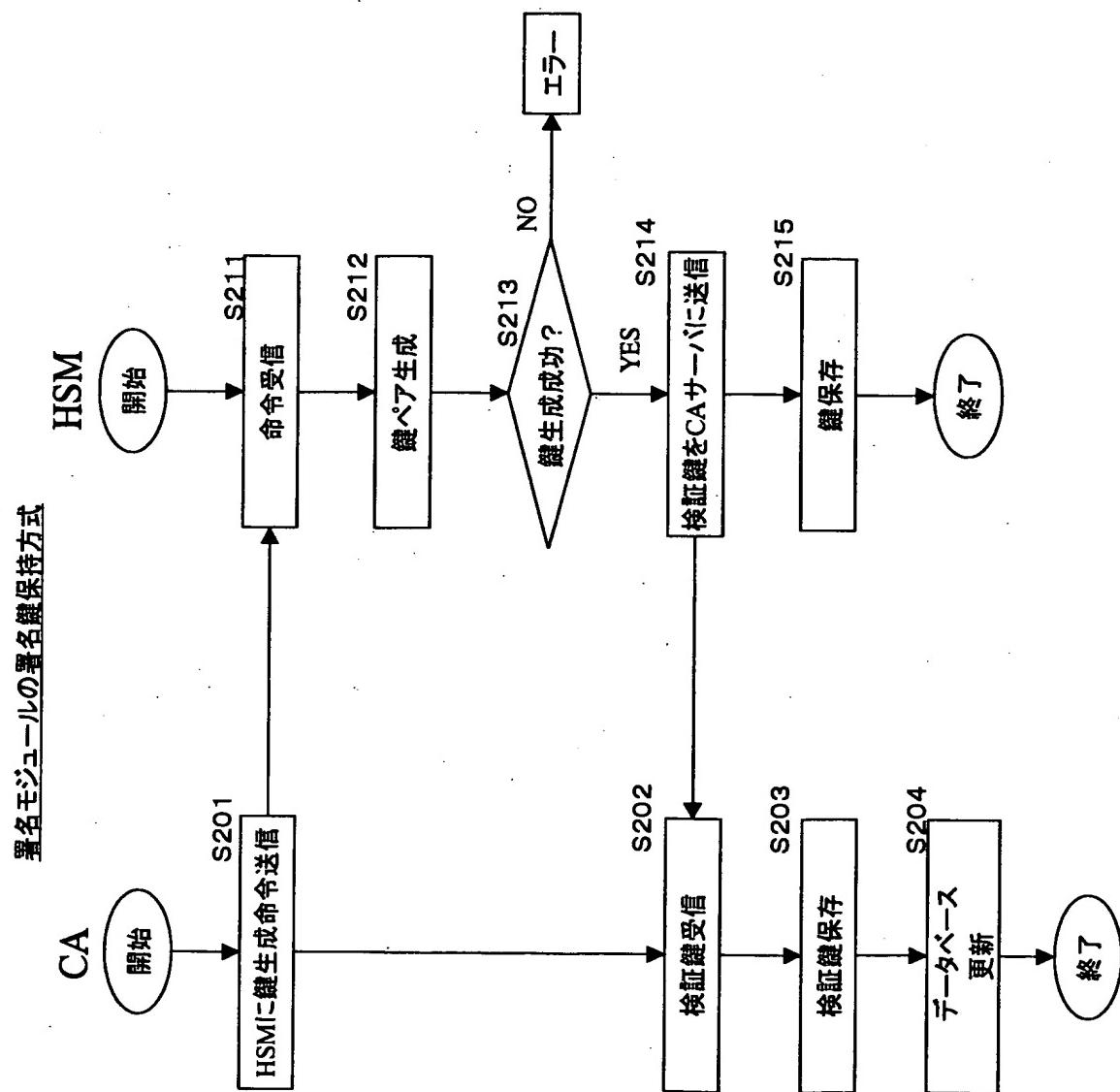
【図17】

同一署名鍵を複数署名モジュールで共有する方式

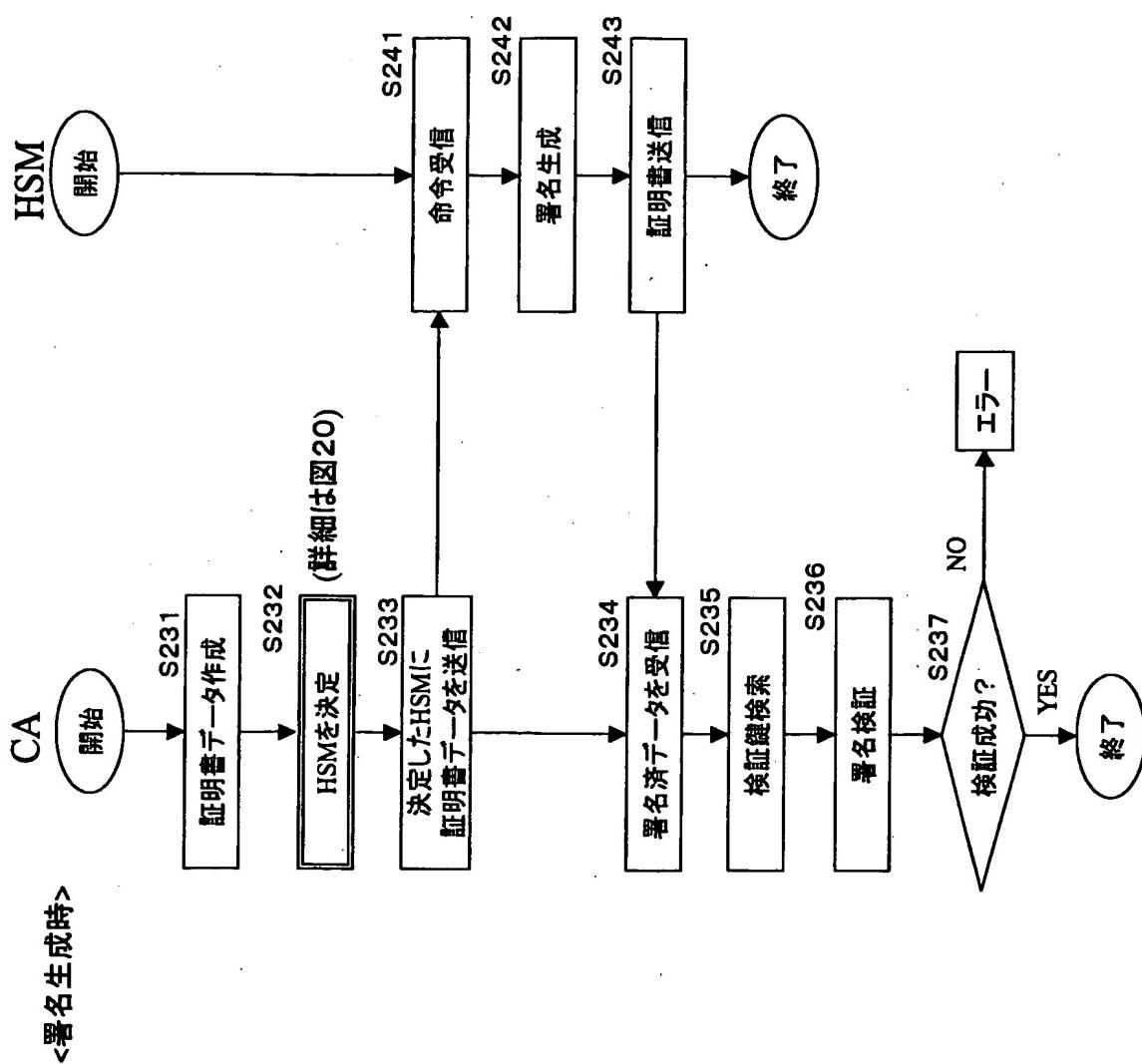
[署名モジュール2~N]



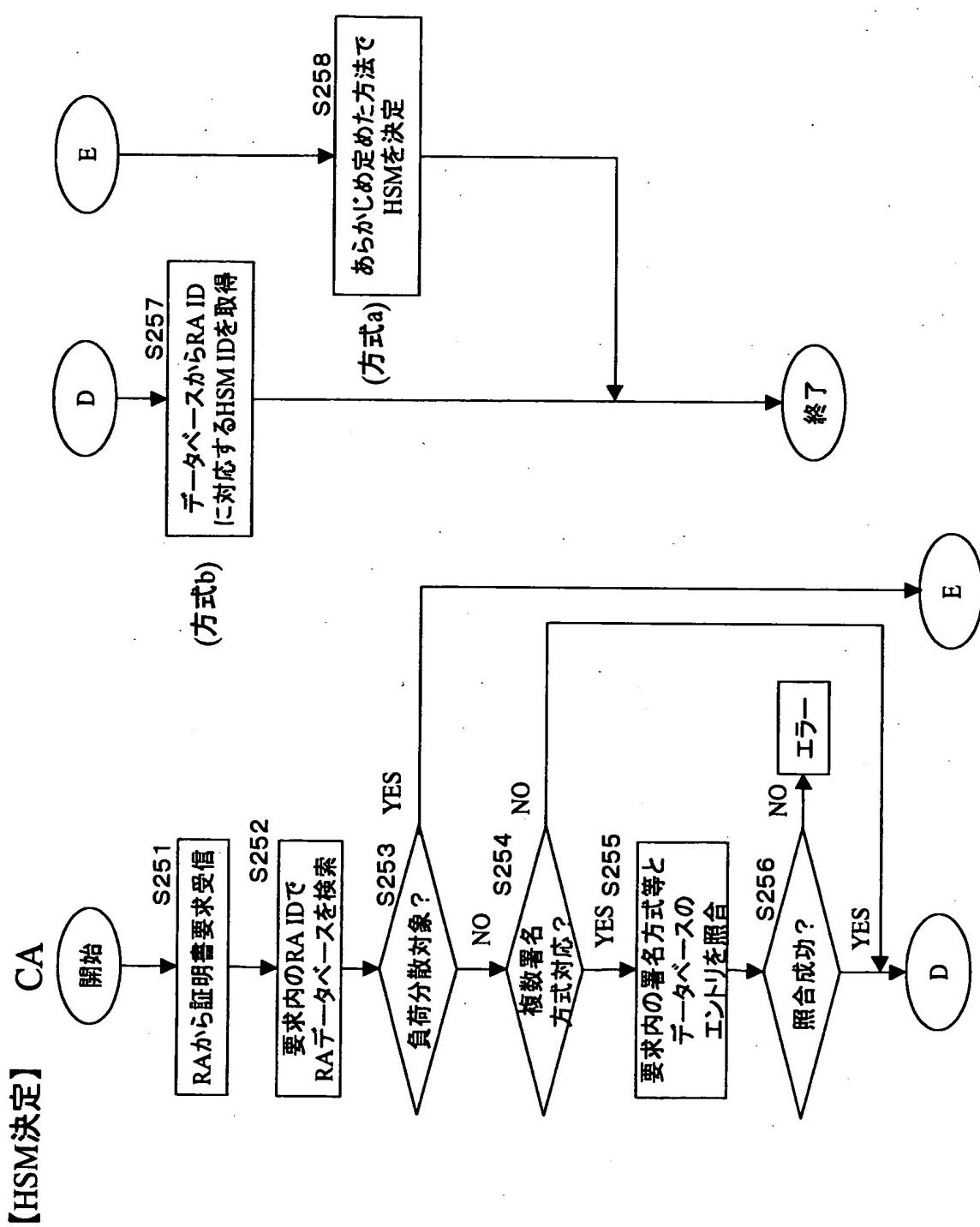
【図18】



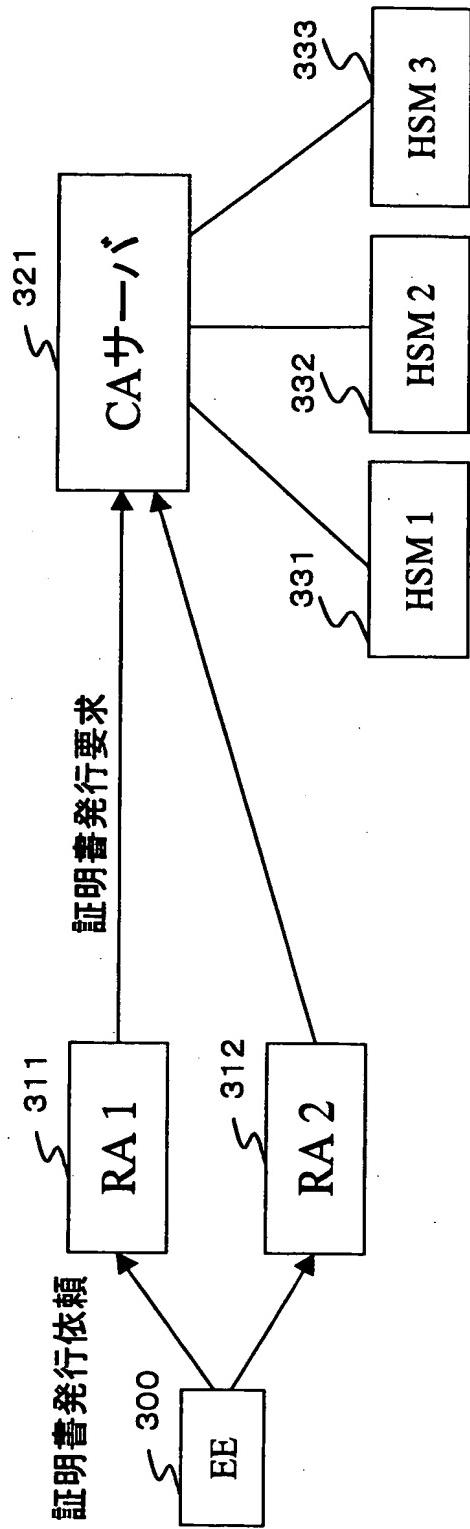
【図19】



【図20】



【図21】



RA ID	複数署名 方式利用	署名方式	鍵長	パラメータ	負荷分散	利用HSM
RA1	X	RSA	1024 bit	-	X	HSM 1
RA2	O	RSA	2048 bit	-	X	HSM 2
RA2	O	ECDSA	192 bit	p=XX,...	X	HSM 3
RA2	O	ECDSA	192 bit	p=YY,...	X	HSM 3

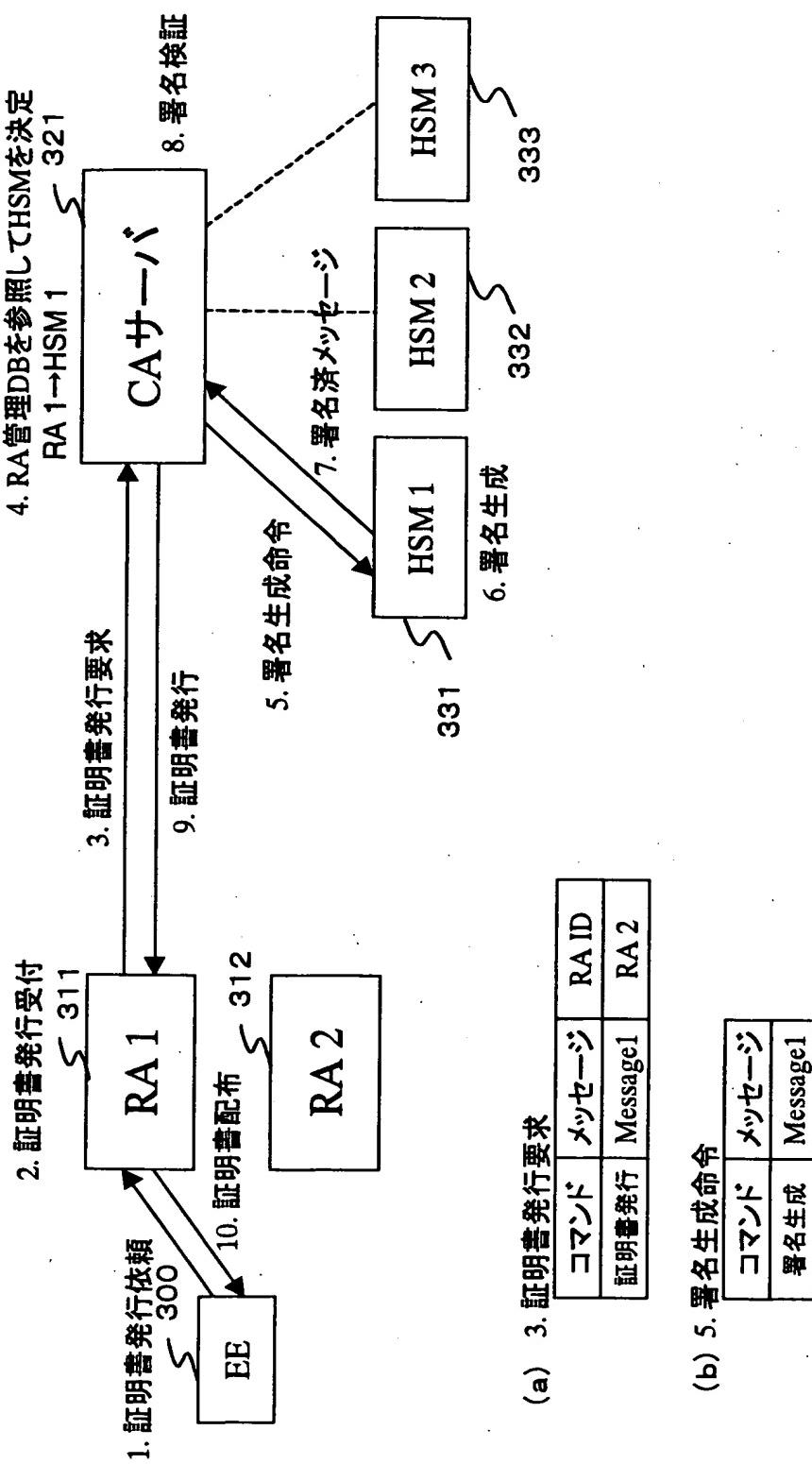
(a) RA管理データベース

HSM ID	署名方式	鍵長	パラメータ	検証鍵
HSM 1	RSA	1024 bit	-	◇TT
HSM 2	RSA	2048 bit	-	◆TT
HSM 3	ECDSA	192 bit	p=XX,...	△TT
"	ECDSA	192 bit	p=YY,...	▲TT

(b) 検証鍵データベース

【図22】

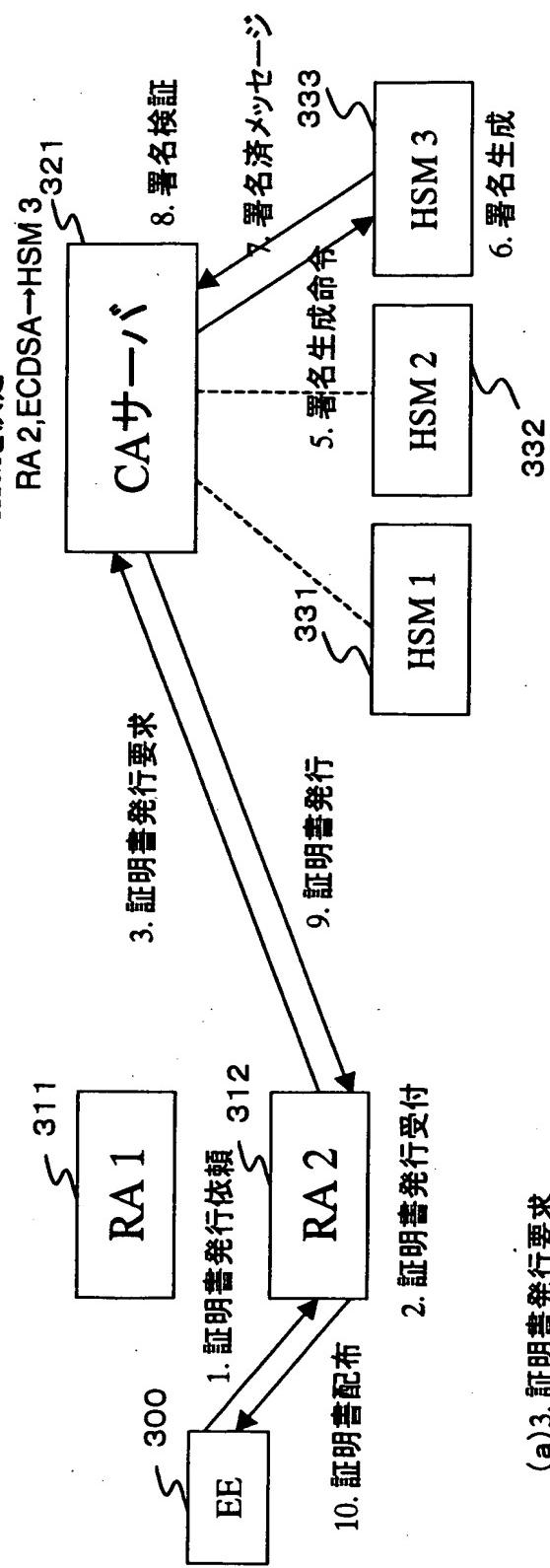
複数署名方式対応
1) 署名選方式固定



【図23】

複数署名方式対応
2) 署名選択方式選択

4. 証明書発行要求とRA管理DBから
HSMを選択



(a)3. 証明書発行要求

コマンド	メッセージ	RA ID	署名方式	鍵長	パラメータ
証明書発行	Message2	RA 2	ECDSA	192 bit	$p=XX, \dots$

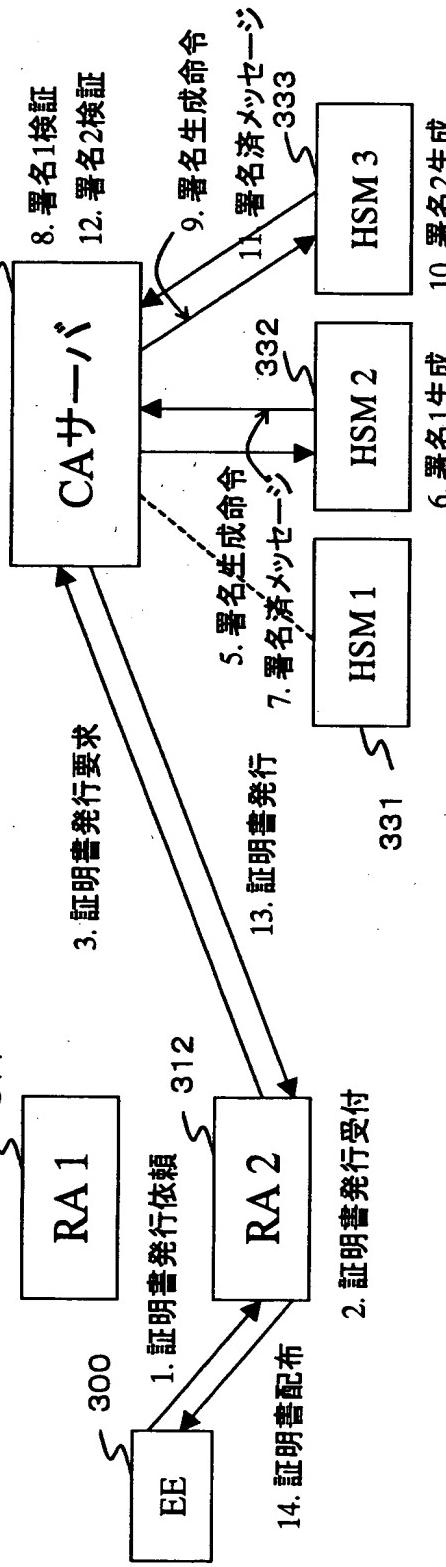
(b)5. 署名生成命令

コマンド	メッセージ	鍵長	パラメータ
署名生成	Message2	192 bit	$p=XX, \dots$

複数署名方式対応
3) 署名選択、同時に複数の署名付け

4. 証明書発行要求とRA管理DBから
HSMを決定

RA2,RSA→HSM2
RA2,ECDSA→HSM3
3-321



(a) 3. 証明書発行要求

コマンド	メッセージ	RA ID	署名方式1	鍵長	署名方式2	鍵長	パラメータ
証明書発行	Message3	RA 2	RSA	2048 bit	ECDSA	192 bit	$p=YY, \dots$

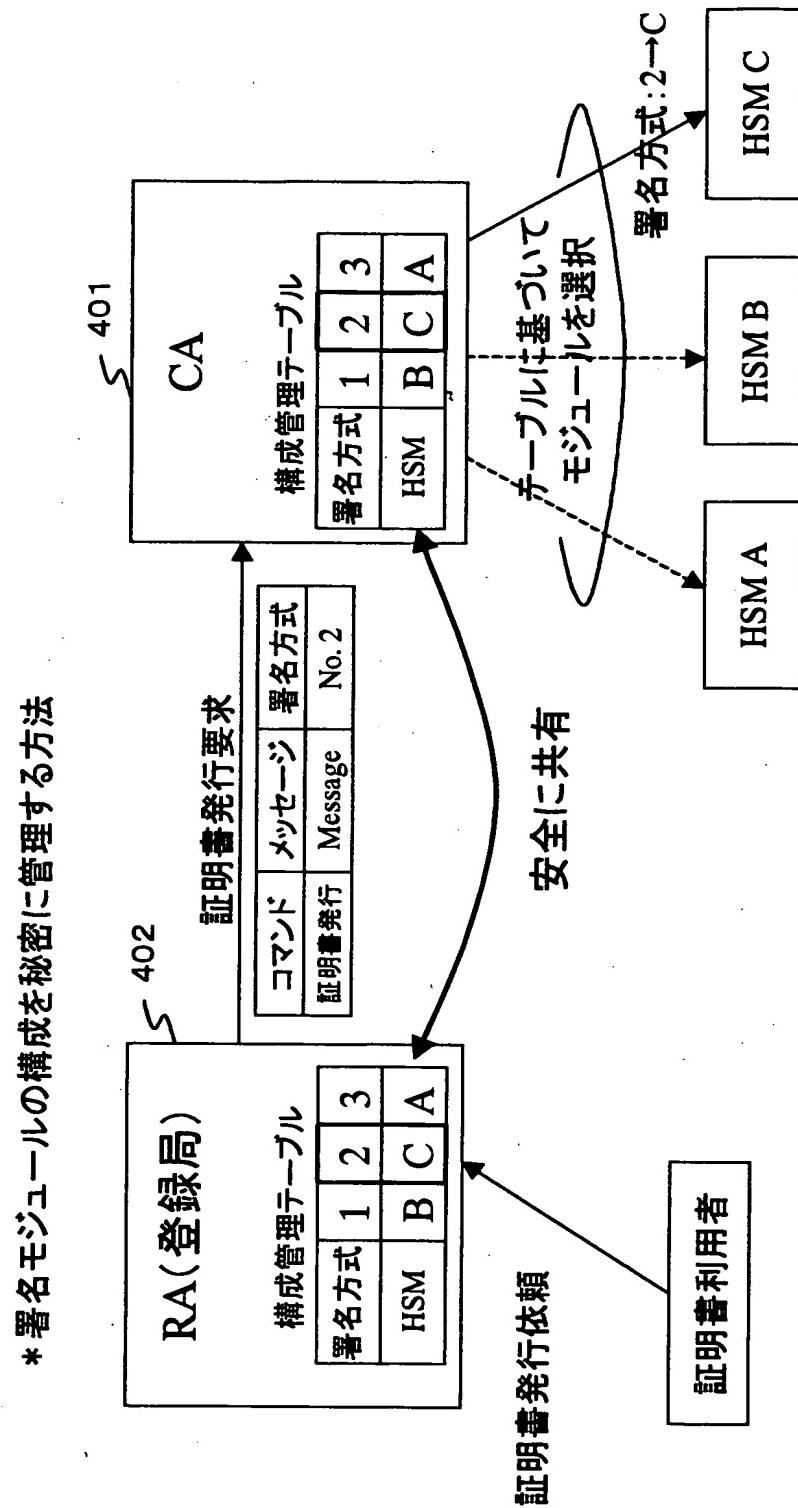
(b) 5. 署名生成命令

コマンド	メッセージ	鍵長
署名生成	Message3	2048 bit

(c) 9. 署名生成命令

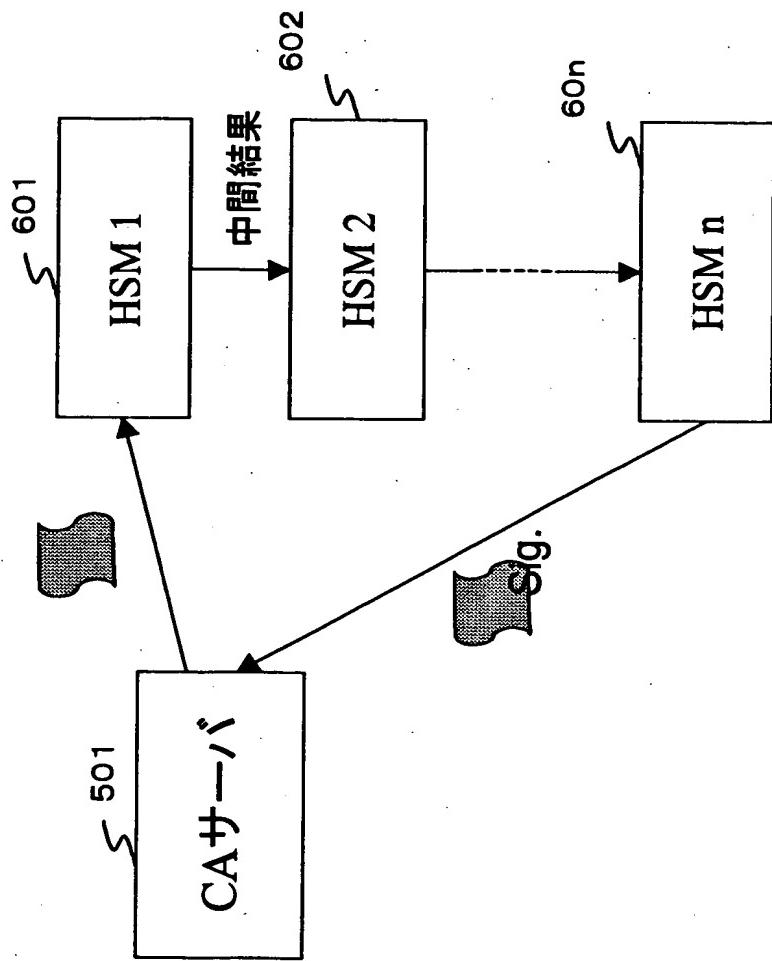
コマンド	メッセージ	鍵長	パラメータ
署名生成	Message3	192 bit	$p=YY, \dots$

【図25】



【図26】

*複数の署名モジュールを連携して利用する方法



【書類名】 要約書

【要約】

【課題】 複数の署名方式を適用した公開鍵証明書を1つの認証局において発行可能とした構成を提供する。

【解決手段】 認証局（CA）が、RSA、ECCなど様々な署名方式を実行する複数の署名モジュールを持ち、複数の署名方式を選択して実行可能とした。認証局（CA）は、登録局（RA）に対応する署名方式に従って署名方式を選択して実行する。従って、特定の署名方式を実行する認証局を署名方式に応じて複数構成する必要がなく、1つの認証局のみで、様々な署名方式を要求する複数の登録局（RA）に応じて、各種の署名が可能となり、1つの認証局のみで複数の署名方式の異なる公開鍵証明書を発行することが可能となる。また、1つの公開鍵証明書に複数の異なる方式による署名を付加する処理も可能となる。

【選択図】 図3

出願人履歴情報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社



Creation date: 10-21-2004
Indexing Officer: MALI3 - MOHAMED ALI
Team: OIPEBackFileIndexing
Dossier: 10041964

Legal Date: 02-14-2002

No.	Doccode	Number of pages
1	CTMS	1

Total number of pages: 1

Remarks:

Order of re-scan issued on